Definition and Use of Software Requirement Patterns in Requirements Engineering

PhD thesis of

Cristina Palomares

Supervised by

Xavier Franch and **Carme Quer**

Definition and Use of Software Requirement Patterns in Requirements Engineering

PhD thesis

Presented by:

Cristina Palomares Bonache

Advisors:

Dr. Xavier Franch

Dr. Carme Quer

Programa de Doctorat en Computació
Departament de Ciències de la Computació
Universitat Politècnica de Catalunya

July 2016

"Para mis abuelos, Noni y Santos. Me visteis empezar este viaje, pero estáis viendo el final desde el cielo. Estoy segura que hoy estaríais muy orgullosos. Gracias por creer siempre en mí."

"To my grandparents, Noni and Santos. You saw me starting this journey, but you are seeing the end from heaven. I am sure you would be very proud today. Thanks for always believing in me."

Acknowledgments

I did not know where I was putting myself into when I started this journey. As most PhD candidates, I guess. It has been hard as hell, with lots of ups and downs, but I admit it has been worth it. I have learnt tons of different things, and I have got over limits I wouldn't think I would be able to get over. All my effort, pain and suffer of the last four years is reflected today in this thesis. However, I wouldn't have reached the end without the support of many people.

First of all, I would like to thank my PhD advisors, Professors Carme Quer and Xavier Franch, who have been extremely helpful and have given all the assistance, support, and guidance I could ask for. Thanks to Professor Tony Gorschek too, who made my research stay in Sweden a lot richer.

I would also like to acknowledge my colleagues in the office, David Ameller, Lidia López, Silverio Martínez, Marc Oriol, and Xavier Oriol, and all the GESSI members, who have helped me through all these years, and with whom I have shared many moments.

Last, but not least, a very special thanks to my family and partner, who even not understanding what I was doing in this journey and why I was doing it, have always been on my side and have supported me no matter what. Thanks to mom and dad, Antonia and Manuel, to my sister and brother-in-law, Toñi and Raúl, to my nephews (in case they read that sometime in the future), Eric and Álex, and to my other half, my very special you. I could thank all of you for a million things, but the most important one is you being able to make me disconnect when I needed it, and take a smile out of me even when my mood was not the best. At the end, small details and good times, even seeming foolish, are what is really appreciated. I couldn't have done it without you.

This work has been possible thanks to the funding of the Spanish grant AP2010-4414 and the Spanish projects TIN2010-19130-C02-01 and TIN2013-44641-P.

Agradecimientos

No tenía ni idea de donde me estaba metiendo a mí misma cuando empecé con esto del doctorado. Como muchos de los otros estudiantes de doctorado, supongo. Ha sido duro, muy duro, con muchos altos y bajos por el camino, pero tengo que admitir que ha valido la pena. He aprendido millones de cosas, y he superado límites que pensaba que no podría superar. Todo el esfuerzo, dolor y sufrimiento de los últimos cuatro años se ve reflejado hoy en esta tesis. Sin embargo, no habría llegado al final sin el soporte que mucha gente me ha brindado por el camino.

Primero de todo, quería agradecer a mis directores de tesis, los doctores Carme Quer y Xavier Franch, quienes han sido de mucha ayuda y me han dado todo el apoyo, soporte y orientación que he necesitado. Gracias también al doctor Tony Gorschek, quién hizo mi estancia de investigación en Suecia mucho más rica.

También quería agradecer a mis compañeros de oficina, David Ameller, Lidia López, Silverio Martínez, Marc Oriol, y Xavier Oriol, y a todos los miembros de GESSI, quienes me han ayudado a lo largo de estos años, y con quienes he compartido muchos momentos.

Por último, pero no por ello menos importantes, un gracias muy muy especial a mi familia y pareja, quienes a pesar de no entender que estaba haciendo en esto del doctorado y porque lo la estaba haciendo, siempre han estado a mi lado y me han apoyado sin importar el qué. Gracias a mi padre y a mi madre, Manuel y Antonia, a mi hermana y a mi cuñado, Toñi y Raúl, a mis sobrinos (en caso de que algún día lleguen a leer esto), Eric y Álex, y a mi otra mitad, ese tú tan especial. Os podría agradecer un millón de cosas, pero la más importante es que habéis sido capaces de hacerme desconectar cuando lo necesitaba, y me habéis sacado una sonrisa incluso cuando mi humor no era el mejor. Al final, los pequeños detalles y los buenos ratos, aunque parezcan tonterías, son lo que más se aprecian. No podría haber hecho esto sin vosotros.

Este trabajo ha sido posible gracias al financiamiento de la beca española AP2010-4414 y de los proyectos españoles TIN2010-19130-C02-01 y TIN2013-44641-P.

Abstract

The final quality of software products and services depends on the requirements stated in the Software Requirements Specifications (SRSs). However, some problems like ambiguity, incompleteness and inconsistency have been reported in the writing of SRSs, especially when natural language is used. Requirements reuse has been proposed as a key asset for requirements engineers to efficiently elicit, validate and document software requirements and, as a consequence, obtain SRSs of better quality through more effective engineering processes.

Among all the possible techniques to achieve reuse, patterns hold a prominent position. In their most classical form, patterns describe problems that occur over and over again, and then describe the core of the solution to these problems. Software engineering practitioners have adopted the notion of pattern in several contexts, remarkably related to software design (e.g. design patterns and software architectural patterns), but also in other software development phases, both earlier and later. Following this strategy, requirement patterns emerge as a natural way to reuse knowledge during the Requirements Engineering (RE) stage.

Although there have been several techniques proposed to reuse requirements, it has been observed that no concrete proposal has achieved a wide acceptance, neither any covered all the necessary elements to encourage organizations to adopt requirements reuse. As a consequence, this thesis proposes the use of Software Requirement Patterns (SRPs) as a means to capture and reuse requirements knowledge in the context of information technology projects. Following the typical context-problem-solution structure of patterns, an SRP mainly consists of: a template (solution) that may generate one or more requirements when applied in a certain project, and some information (context-problem) to identify its applicability in that project. To facilitate their use, SRPs are encapsulated inside the PABRE (PAttern-Based Requirements Elicitation) framework. The framework covers all the elements that could be critical for the adoption of a requirements reuse technique. Specifically, the framework includes:

- A metamodel that describes the structure and semantics of SRPs and their organization inside a catalogue.

- An SRP catalogue composed by non-functional, non-technical and functional SRPs, the functional ones being specific for the content management system domain.
- A method for guiding the use of an SRP catalogue during requirements elicitation and specification, as well as another one for constructing and updating it.
- An economic model to perform cost-benefit analysis on the adoption of SRPs based on return-on-investment.
- The PABRE system as technological support.

In order to analyse the benefits and drawbacks of the SRPs proposed in this thesis, two empirical studies have been carried out to investigate the perception of participants about requirement patterns in general and SRPs in particular. The first one is an exploratory survey addressed to information technology people with industrial experience in RE, which analyses the current state of the practice of requirement patterns approaches. The second one corresponds to a set of semi-structured interviews, focused on the SRP approach, conducted to requirements engineers of Swedish organizations. Moreover, as it has been discovered that there are few empirical studies showing the state of the practice of requirements reuse in industry, the first study also explores the current situation of requirements reuse practices in organizations.

Table of Contents

Acknowledgments i

Abstract v

Table of Contents vii

List of Figures xiii

List of Tables xvii

Introduction

1 Introduction 1

1.1 Context and Terminology 1

1.1.1 Requirements Engineering 1

1.1.2 Requirements Reuse 3

1.1.3 Reuse through Patterns 5

1.2 Motivation 6

1.3 Research Questions 8

1.4 Methodological Approach 9

1.5 Research Context and Contributions 11

1.5.1 Use of Patterns for Reusing Knowledge during RE 12

1.5.2 Definition of the SRP Structure 13

1.5.3 Integration of SRPs in RE processes 14

1.5.4 Empirical Research about Requirement Reuse and Requirement Patterns 15

1.6 List of Publications 15

1.7 Structure of this Thesis 17

PART I - The PABRE Framework

2 Requirements Reuse and Patterns: State of the Art and the Practice ... 21

2.1 Introduction 21

2.2 State of the Art: Patterns for Reuse on RE 22

2.2.1 Systematic Mapping Studies ... 22
2.2.2 Planning the SMAP ... 23
2.2.2.1 Identification of the Need for a Study ... 24
2.2.2.2 Scoping the Study ... 24
2.2.2.3 Defining the Protocol of the Study ... 25
2.2.3 Conducting the SMAP ... 28
2.2.3.1 Selection of Primary Studies ... 28
2.2.3.2 Data Extraction ... 30
2.2.3.3 Analysis and Classification Process ... 31
2.2.4 Reporting the SMAP ... 32
2.2.4.1 Results ... 32
2.2.4.2 Threats to Validity ... 57
2.2.5 Conclusions of the Study ... 60
2.3 State of the Practice: Requirements Reuse and Patterns ... 68
2.3.1 Background on Requirements Reuse Empirical Studies ... 70
2.3.2 Research Approach ... 75
2.3.3 Characterization of the Respondents ... 79
2.3.4 On Requirements Reuse Adoption ... 83
2.3.5 On Barriers to the Adoption of Requirements Reuse ... 87
2.3.6 On Requirements Reuse through Patterns ... 88
2.3.7 Discussion of the Results ... 92
2.3.7.1 About the Current State of the Practice of Requirements Reuse ... 92
2.3.7.2 Reasons behind the Lack of Adoption of Existing Requirements Reuse Proposals ... 95
2.3.7.3 About the Benefits and Drawbacks of Using a Catalogue of RPs ... 96
2.3.8 Threats to Validity ... 98
2.3.9 Conclusions of the Study ... 101
2.4 General Conclusion ... 103
3 Overview of the PABRE Framework ... 105
3.1 What is PABRE ... 105

3.2 SRP in PABRE **106**

3.3 SRP Catalogue in PABRE **110**

3.4 Possible Benefits of PABRE **111**

3.5 Possible Drawbacks of PABRE **115**

4 Metamodel of an SRP Catalogue **119**

4.1 Introduction **119**

4.2 Core Structure of SRPs **120**

4.2.1 Analysis of SRS Documents 120

4.2.2 Metamodel of SRP Structure 131

4.3 Relationships among SRPs **134**

4.3.1 Analysis of SRS Documents 134

4.3.2 Metamodel of SRP Relationships 138

4.4 Classification of SRPs **139**

4.4.1 Analysis of SRS Documents 139

4.4.2 Metamodel of SRP Classification Schemas 142

5 Use of an SRP Catalogue **145**

5.1 Introduction **145**

5.2 Identification of SRPs **146**

5.2.1 Browsing/Searching the SRP Catalogue 146

5.2.2 Alternatives for Identifying SRPs 151

5.3 Application of SRP **152**

6 Lifecycle of an SRP Catalogue **159**

6.1 Introduction **159**

6.2 Creation of an SRP Catalogue **160**

6.3 Update of an SRP Catalogue **173**

7 Catalogue of SRPs **177**

7.1 Introduction **177**

7.2 Non-Functional SRP Catalogue **178**

7.3 Non-Technical SRP Catalogue **182**

7.4 Functional SRP Catalogue **188**

8 Return-On-Investment of an SRP Catalogue 195
8.1 Introduction 195
8.2 Background 196
8.3 Definition of the ROI Model for SRPs..................................... 198
8.4 Discussion 201
9 The PABRE System 205
9.1 Architecture 205
9.2 PABRE System for the Requirements Elicitation and Specification 209
9.3 PABRE System for the Creation and Update of an SRP Catalogue 215

PART II - Validation of the PABRE Framework

10 Validation of the SRP Metamodel 225
10.1 Introduction 225
10.2 Analysis with Respect to NTRs and FRs..................................... 226
10.3 Analysis with Respect to Quality 231
10.4 Conclusions 235
11 Validation of the PABRE Framework 237
11.1 Introduction 237
11.2 Research Approach 238
11.2.1 Research Question..................................... 238
11.2.2 Research Method..................................... 239
11.2.3 Protocol and Research Team 240
11.2.4 Sampling..................................... 240
11.2.5 Procedure and Instruments..................................... 240
11.2.6 Data Collection..................................... 241
11.2.7 Data Analysis Procedure..................................... 242
11.3 Characterization of the Participants, Organizations and Projects .. 244
11.4 Results..................................... 248

11.5 Discussion of the Results 261
11.6 Threats to Validity 264
11.7 Conclusions 267

Conclusions and Future Work

12 Conclusions and Future Work 273
12.1 Conclusions 273
12.2 Comparison with Other Approaches 276
12.3 Future Work 282

List of Abbreviations 287
Appendix I – SMAP Need Identification Works 289
Appendix II – SMAP Data Extracted 291
Appendix III – SRP Metamodel 319
Appendix IV – Interview Guide 320
Appendix V – Interviews Code Relationships 327
SMAP Bibliography 329
Bibliography 337

List of Figures

Figure 2.1 – Selection process of the systematic mapping........30
Figure 2.2 – Primary studies per year........34
Figure 2.3 – Primary studies per venue types........34
Figure 2.4 – Primary studies per continents of the authors........35
Figure 2.5 – Proposals per type of proposal........37
Figure 2.6 – Proposals per scope........37
Figure 2.7 – Proposals per domain........38
Figure 2.8 – Proposals per type of requirements being reused........40
Figure 2.9 – Proposals per artefact to reuse........42
Figure 2.10 – Proposals per notation used to define the artefact to reuse........44
Figure 2.11 – Proposals per structure in which the artefact to reuse is contained........46
Figure 2.12 – Proposals per information accompanying the artefact to reuse........47
Figure 2.13 – Proposals per relationships among artefacts to reuse........48
Figure 2.14 – Proposals per arrangement of the artefacts to reuse........48
Figure 2.15 – Proposals per specific classification proposed for the artefacts to reuse........49
Figure 2.16 – Proposals per metamodel for the artefact to reuse........49
Figure 2.17 – Proposals per metamodel for the arrangement of the artefacts to reuse........49
Figure 2.18 – Proposals per method to construct the artefact to reuse........50
Figure 2.19 – Proposals per method to reuse the artefacts........50
Figure 2.20 – Proposals per method to update the catalogue of artefacts to reuse..51
Figure 2.21 – Proposals per existence of a catalogue of artefacts to reuse........51
Figure 2.22 – Proposals per domain of the catalogue of artefacts to reuse........52
Figure 2.23 – Proposals per state of the catalogue of artefacts to reuse........52
Figure 2.24 – Proposals per existence of tools........54
Figure 2.25 – Proposals per tests done........54
Figure 2.26 – Proposals per type of research........56
Figure 2.27 – Distribution of responses for the level of industrial experience related to RE........79
Figure 2.28 – Global view of survey participant locations........80
Figure 2.29 – Distribution of responses for educational background........80
Figure 2.30 – Distribution of responses for years of experience........80

Figure 2.31 – Distribution of responses for organization size 81
Figure 2.32 – Distribution of responses for languages used to specify requirements ...82
Figure 2.33 – Distribution of responses for requirements elicitation methods 83
Figure 2.34 – Distribution of responses for requirements reuse level.....................84
Figure 2.35 – Distribution of responses for requirements reuse techniques..........85
Figure 2.36 – Distribution of responses for problems that hinder the adoption of requirements reuse proposals..88
Figure 3.1 – SRPs in the PABRE framework...105
Figure 3.2 – PABRE framework overview..106
Figure 3.3 – Users Capacity SRP 107
Figure 3.4 – SRP classification schemas...111
Figure 4.1 – Number of SRS documents per concept related to reusable requirements (1) 122
Figure 4.2 – Number of SRS documents per concept related to non-reusable requirements (2) 122
Figure 4.3 – Number of requirements per type of entity 124
Figure 4.4 – Number of clusters in the SRS documents per number of requirements per cluster 125
Figure 4.5 – Number of clusters in the SRS documents with more than one requirement 125
Figure 4.6 – Concepts represented by more than one cluster in the same SRS.....127
Figure 4.7 – Metamodel: basic structure of an SRP 131
Figure 4.8 – Relationships found in SRS documents distributed per granularity and type 137
Figure 4.9 – Metamodel: SRP relationships...138
Figure 4.10 – Excerpt of the relationships found among NF-SRPs 139
Figure 4.11 – Number of classifiers per SRS document 141
Figure 4.12 – Classifiers containing more than one concept in the same SRS......141
Figure 4.13 – Metamodel: SRP classification schemas 143
Figure 5.1 – Overview of the PABRE use methods..146
Figure 5.2 – Activities followed during the SRP identification with PABRE...........147
Figure 5.3 – Excerpt of the ISO/IEC 25010 classification schema...........................149
Figure 5.4 – Requirements extraction in PABRE...152
Figure 5.5 – Activities followed during the requirements extraction with PABRE ...153
Figure 5.6 – Sample of requirements extracted with PABRE using the Users Capacity SRP...154
Figure 6.1 – Lifecycle of an SRP catalogue 159

Figure 6.2 – Overview of the SRP creation method 161
Figure 6.3 – Activities carried out during the SRP creation method....................162
Figure 6.4 – Sample of requirements extracted from SRSs 162
Figure 6.5 – Feedback information and actions taken for the update of the SRP catalogue 174
Figure 9.1 – The PABRE system 206
Figure 9.2 – Use case diagram of the PABRE system....................206
Figure 9.3 – PABRE subsystems architecture and technologies 207
Figure 9.4 – Using the PABRE system before starting the elicitation and specification of requirements 210
Figure 9.5 – Entering IT project information with PABRE-Proj....................210
Figure 9.6 – Using the PABRE system for applying an SRP....................211
Figure 9.7 – Creating a requirement as an application of SRP in PABRE-Proj......212
Figure 9.8 – Giving values to the parameters of an SRP application requirement in PABRE-Proj 212
Figure 9.9 – Browsing the elicited and specified requirements in PABRE-Proj 213
Figure 9.10 – Main scenarios while eliciting and specifying requirements with the PABRE system....................213
Figure 9.11 – Using the PABRE system for generating the SRS....................214
Figure 9.12 – Project export functionalities in PABRE-Proj 214
Figure 9.13 – PDF generated with PABRE-Proj corresponding to the SRS...........215
Figure 9.14 – Using the PABRE system to create an SRP 216
Figure 9.15 – Useful information while creating an SRP in PABRE-Man................217
Figure 9.16 – Editing an SRP template with PABRE-Man 217
Figure 9.17 – Editing a parameter with PABRE-Man 218
Figure 9.18 – Using the PABRE system to generate the report about the SRP data of use....................219
Figure 9.19 – Using the PABRE system to update the SRP catalogue....................219
Figure 9.20 – Browsing the requirements associated to an SRP and its data of use with PABRE-Man....................221

List of Tables

Table 1.1 – Shaw's list of research settings 10
Table 1.2 – Shaw's list of research products 10
Table 1.3 – Shaw's list of validation techniques 10
Table 1.4 – List of accepted publications 16
Table 1.5 – List of publications in progress 16
Table 1.6 – Summary of chapters of this dissertation 17
Table 2.1 – Research question 1 and its subquestions 21
Table 2.2 – Research question 1.1 and its subquestions 25
Table 2.3 – Keywords related to the Population and Intervention 26
Table 2.4 – Data extraction form 30
Table 2.5 – Most targeted venues by the primary studies 34
Table 2.6 – Countries with the biggest figure of primary studies 35
Table 2.7 – Scope by type of proposal 38
Table 2.8 – Domain by type of proposals 40
Table 2.9 – Requirement types by type of proposals 41
Table 2.10 – Artefact to reuse by type of proposals 43
Table 2.11 – Notation language by type of proposals 45
Table 2.12 – Structure in which the artefact to reuse is contained by type of proposal 46
Table 2.13 – Existence of a catalogue of artefacts to reuse by type of proposal 51
Table 2.14 – State of the catalogue of artefacts to reuse by its domain 52
Table 2.15 – Existence of tools by type of proposal 54
Table 2.16 – Tests done by type of proposal 55
Table 2.17 – Tests done by existence of tools 55
Table 2.18 – Type of research by type of proposal 56
Table 2.19 – Type of research by tests done 56
Table 2.20 – Details of the most consolidated SMAP proposals 68
Table 2.21 – Related work on industrial case studies on requirements reuse 70
Table 2.22 – Related work on surveys and interviews on RE in general that somehow address requirements reuse 72
Table 2.23 – Related work on surveys and interviews on requirements reuse 74
Table 2.24 – Research question 1.2 and its subquestions 76
Table 2.25 – Relation between survey sections and subresearch questions 77
Table 2.26 – Distribution of responses for project domain 81

Table 2.27 – Cross-tabulation of requirements reuse level and organization size ... 84
Table 2.28 – Cross-tabulation of requirements reuse level and requirements reuse techniques ... 85
Table 2.29 – Average response for requirement types likely to be more similar between projects ... 87
Table 2.30 – Average response for problems mitigated by the use of an RP catalogue ... 90
Table 2.31 – Average response for critical factors influencing the adoption of an RP catalogue ... 90
Table 2.32 – Average response for barriers influencing the adoption of an RP catalogue ... 91
Table 3.1 – ISO/IEC/IEEE 29148 good characteristics for requirements addressed by PABRE ... 113
Table 4.1 – Excerpt of requirements found in the analysed SRS documents (1) ... 121
Table 4.2 – Excerpt of requirements found in the analysed SRS documents (2) ... 128
Table 4.3 – Users Capacity SRP ... 133
Table 4.4 – Excerpt of requirements found in the analysed SRS documents (3) ... 137
Table 4.5 – Excerpt of the ISO/IEC 25010 based classification schema with NF-SRPs classified ... 143
Table 6.1 – Sample of requirements after Requirements Alignment is finished ... 163
Table 6.2 – Example of new SRP candidate ... 165
Table 6.3 – Example of associated SRP candidate that does not imply the addition of a new candidate ... 166
Table 6.4 – Example of associated SRP candidate that implies the addition of a new candidate ... 166
Table 6.5 – Sample of SRP candidates after Requirements Analysis is finished ... 166
Table 6.6 – Example of SRP candidate appearing in one SRS ... 167
Table 6.7 – Sample of SRP abstractions during SRP Formulation (1) ... 168
Table 6.8 – Sample of SRP abstractions during SRP Formulation (2) ... 169
Table 7.1 – NF-SRP catalogue classified using the ISO/IEC 25010-based classification schema ... 178
Table 7.2 – Number of relationships of the NF-SRP catalogue classified per type and elements ... 180
Table 7.3 – Failure Alerts NF-SRP ... 181
Table 7.4 – NT-SRP catalogue classified using the ISO/IEC 25010-based classification schema ... 184
Table 7.5 – Number of relationships of the NT-SRP catalogue classified per type and elements ... 185

Table 7.6 – Supplier Economic Information NT-SRP 186
Table 7.7 – F-SRP catalogue classified using the ISO/IEC 25010-based classification schema 189
Table 7.8 – Number of relationships of the F-SRP catalogue classified per type and elements 191
Table 7.9 – Content Version Management F-SRP 193
Table 8.1 – Cost-benefit factors to calculate the ROI of adopting an SRP catalogue in an organization 200
Table 8.2 – Basic parameters in order to feed the factors of Table 8.1 201
Table 10.1 – Results of observations made on NTRs and FRs 227
Table 10.2 – Metamodel quality features from López et al. used for the SRP metamodel quality analysis 232
Table 10.3 – Metamodel quality features from Bertoa's et al. Framework used for the SRP metamodel quality analysis 234
Table 11.1 – Research question of the interview-based empirical study and its subquestions 238
Table 11.2 – Characteristics of participants 246
Table 11.3 – Characteristics of organizations 246
Table 11.4 – Characteristics of projects 247
Table 11.5 – Categories of the responses to whether SRPs would enable reuse 249
Table 11.6 – Categories of the reasons that explain that SRPs would enable reuse 249
Table 11.7 – Categories related to how SRPs could be used in the organization 250
Table 11.8 – Categories of the responses to whether SRPs would be relevant for all types of requirements 252
Table 11.9 – Categories of types of requirements for which SRPs would be or would not be relevant 252
Table 11.10 – Categories of the pros that would come from using SRPs 254
Table 11.11 – Categories of the cons that would come from using SRPs 256
Table 11.12 – Categories of the introduction factors that could help during the introduction of SRPs 258
Table 11.13 – Categories of the barriers that could hinder the introduction of SRPs 260
Table 11.14 – Mapping among the possible drawbacks of the PABRE framework and the cons and barriers identified in the interviews 269
Table 12.1 – Comparison of the PABRE framework with other approaches (1/3) .. 277
Table 12.2 – Comparison of the PABRE framework with other approaches (2/3) .. 278
Table 12.3 – Comparison of the PABRE framework with other approaches (3/3) .. 279

Definition and Use of Software Requirement Patterns in Requirements Engineering

CHAPTER 1

Introduction

1.1 Context and Terminology

The present thesis has grown around three software engineering topics: Requirements Engineering (RE) (as the thesis deals with the requirements elicitation and specification RE activities), Requirements Reuse (because the thesis proposes a framework to reuse requirements), and Patterns (since the proposed framework uses patterns as the main asset to reuse requirements). These three topics are further explained in the following subsections in order to provide a general context and clarify the concepts and terminology used along this thesis.

1.1.1 Requirements Engineering

A requirement is a singular need that a particular design, product or process must be able to perform. A more complete definition of requirement is given by Hull et al. [1]:

> *"A Requirement is a statement that identifies a product or process operational, functional or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability (by consumers or internal quality assurance guidelines)."*

There are different proposals of requirements types, e.g. the ones proposed by the CMMI standard [2] or SPICE [3]. However, the most common classification, and the one used in this thesis, is the differentiation among functional and non-functional requirements [4], and the classification among technical or non-technical requirements [5]:

- Functional Requirements (FRs) define the functionality that a software system to be implemented shall offer [6], so, by definition, these requirements are technical. An example of FR could be: *The system shall allow to manage versions of the stored documents.*

- Non-Functional Requirements can be technical or non-technical. The technical ones, henceforth Non-Functional Requirements (NFRs), specify system properties, such as environmental and implementation constraints, performance, platform dependencies, maintainability, extensibility, and reliability [7]. NFRs are inherent to the software being produced. An example of NFR could be: *The system shall respond to user interface actions in less than 1 second*. The non-technical ones, henceforth Non-Technical Requirements (NTRs), are those ones that do not refer directly to the intrinsic quality of software, but to the context of the system under analysis; they include economic, political and managerial issues [8]. Therefore, they are triggered by external factors such as organizations, laws, software licensing, and software providers. For illustration, an example of NTR could be: *The system shall be developed in a programming language used by the development team in previous projects*.

NFRs and NTRs are mainly domain-independent, i.e. they appear basically in the same way in different requirement specifications, even if they belong to projects from different IT domains (see for instance the proposals of Withall [9], Suppakkul et al. [10], and Hoffmann et al. [11]). However, that is not the case for FRs. As stated by Lam [12], it is necessary to identify and formalize the reusable requirements for each functional area.

Requirements Engineering (RE) refers to all the activities done around requirements. As defined by Pohl [13]:

> *"Requirements Engineering is the part of software engineering which covers all the activities involved in eliciting, documenting, validating, verifying and managing requirements."*

More specifically: requirements elicitation is the process of acquiring requirements from stakeholders; requirements documentation (also known as requirements specification, which is the term used in this thesis) involves describing adequately the elicited requirements; requirements validation aims to ensure that the requirements documentation is complete, consistent, correct, modifiable and traceable; requirements verification confirms that the designed and built system fully addresses the documented requirements; and requirements management relates to observing the system context to detect system changes and consequently controlling the requirements documentation and the execution of the rest of activities.

The aim of the RE process is to steer the development towards producing the right software [14]. Requirements are the basis for every project, defining the needs of the stakeholders – users, customers, suppliers, developers, and business – on a system-to-be. This is why RE aims at providing an unambiguous, complete and consistent set of requirements that help developing the system.

The final quality of a software system or service greatly depends on the quality of its requirements, which are documented in its Software Requirements Specification (SRS). Eliciting the suitable requirements produces benefits such as preventing errors, improving quality, and reducing risk in Information Technology (IT) projects [15] [16].

1.1.2 Requirements Reuse

Due to the increasing pressure to achieve high quality software in the shortest time possible, different reuse techniques have been introduced for years in the software development process, e.g. the proposals of Biggerstaff and Perlis [17], Schaefer et al. [18], Karlsson [19], Jacobson et al. [20], and Ezran et al. [21]. These techniques facilitate the design and development of components in order to be reused in future IT projects, reducing the development time, improving the system quality and being more competitive on costs.

Ideally, as Prieto-Diaz explains [22], reuse consists on using knowledge in its most abstract form. Requirements represent one of the most abstract level of knowledge in IT projects, so by reusing requirements the level of abstraction of reusable items increases.

It is often the case that, when an organization runs many requirements elicitation processes over time, a significant proportion of requirements is recurrent. As reported by Withall [9], only a fraction of any IT project is specific to the project itself, and the requirements bulk occurs over and over again no matter what the project is for. Requirements reuse is, then, possible, as not all the requirements that define a system are specific for that only system. Capitalizing on knowledge acquired in previous projects seems an adequate strategy to improve the quality of requirements (raising the chances of project success) and the efficiency of the requirements elicitation process.

In a nutshell, requirements reuse can be described as follows:

> *Requirements Reuse refers to taking advantage of requirements knowledge obtained from previous IT projects and later on using this knowledge in a new one.*

Several proposals dealing with requirements reuse have been presented, e.g. Maiden and Sutcliffe [23], Lain [24], and Niazi et al. [25]. The existing proposals examine reuse from different perspectives and apply reuse on different types of specification objects. As explained by Wiegers and Beatty [26], there are different levels of complexity when reusing requirements: "The simplest way to reuse requirements is to copy and paste them from existing specifications. Most sophisticated ways imply reusing entire functional components, relating requirements to design, code, and tests. Numerous reuse options exist in the middle."

The three dimensions of requirements reuse are also described by Wiegers and Beatty [26], namely:

- *The extent of reuse*, i.e. the quantity of material that is being reused. It is possible to reuse just a single requirement, or it is possible to reuse a statement along with any associated attributes, like its rationale, origin, priority, and more, if those are relevant to the target project.
- *The extent of modification*, i.e. how much modification is needed to make existing requirements reusable on the new project. In some cases, a requirement can be reused unchanged, modifying if necessary some of its attributes, such as its priority or rationale, as it applies to the new system. In other cases, an existing requirement is modified to suit exactly the new purpose.
- *The reuse mechanism*, i.e. the mechanism being used to perform the reuse. The most rudimentary form of reuse is simply a copy-and-paste of a piece of requirements information, not retaining a history of where the original information came from, allowing copies to be modified. Another option is reusing existing content by referring to it instead of replicating it, and echoing or not the changes that are done in the master instance. Further possibilities embrace the storage of requirements in a repository; in this case, the capabilities are enormous: keeping older versions of reused requirements, tailoring own versions of requirements without disrupting other copies, etc.

The benefits of effective requirements reuse are stated by Wiegers and Beatty. [26]: "[They] include faster delivery, lower development costs, consistency both within and across applications, higher team productivity, fewer defects, and reduced rework. Reusing trusted requirements can save review time, accelerate the approval cycle, and speed up other project activities, such as testing. Reuse can improve the ability to estimate implementation effort if there is data available from implementing the same requirements on a previous project." However, reuse is not free. It presents risks related to creating the reusable items (as it is necessary to create items with good reuse potential and of high quality), maintaining the reusable items (as the reusable items should be updated so they do not become obsolete) and reusing existing items (as one may end reusing low quality items or more items than needed just because they are available, or taking the reusable items as if they were all the knowledge available and as a consequence ending with less creative systems).

1.1.3 Reuse through Patterns

One of the techniques to achieve reuse is the adoption of patterns. Patterns were proposed by Christopher Alexander in 1977 for the area of architecture and building construction. As Alexander said [27]:

> *"Each pattern describes a problem which occurs over and over again, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."*

Software patterns, thoroughly presented by Schmidt [28], are attempts to describe successful solutions to common software problems. They contain useful models, their design rationale, and the assumptions and constraints that apply when using them. As a consequence, they facilitate reusing and sharing models and design knowledge by allowing their adaptation to fit a specific problem. Usually, these patterns are not invented, but discovered from experience gained in building systems. For many years, research efforts in software development have focused on the identification and use of software patterns.

The first patterns to appear in the software engineering area were design patterns, firstly introduced by Gamma et al. [29], which have as a goal to solve problems identified during the design of software systems. Design patterns incorporate the design of classes, their attributes and associations (jointly with instances, in case they are necessary) that if used as part of a software system are going to solve the stated design problem.

Many other types of patterns have been proposed afterwards. Some of them are: idioms, which are patterns at source code levels [30] that express generally accepted conventions of certain programming languages; software architectural patterns, which propose solutions of the overall structure of software systems depending on characteristics of the system to be built [31]; analysis patterns, which propose object models suitable for common business problems depending on certain circumstances or forces [32]; process patterns, which propose guides to different activities carried out during the software development [33] [34]; and organizational patterns, which propose advices related to organizational pragmatics, i.e. to software developers and users, relationships between them, and relationships between people and software [35] [36].

Finally, patterns have also been proposed for the reuse of knowledge acquired during RE, e.g. the ones introduced by Withall [9], Kotula [37], and Keepence and Mannion [38]. This type of patterns are the ones addressed in this thesis.

1.2 Motivation

The final quality of a software system or service greatly depends on the quality of its requirements, which are documented in its SRS. However, evidence shows that the current state of the practice for acquiring these requirements is still far from being satisfactory [39] [40] [41] [42] [43] [44]. Without a proper set of requirements, any software project has great chances of failure, no matter how well the rest of the project is executed. A study performed by The Standish Group [45] showed that only 9% of large companies and 16% of small companies delivered projects on time and within budget. When the participants were asked for the causes of failed projects, of the top eight factors, five were related to RE, two of which were incomplete requirements (13.1%) and lack of user involvement during RE (12.4%). Besides this report, more recent studies have also identified requirements as being an important risk factor in project failures (like the ones from SwissQ [42], Arnuphaptrairong [46], Galorath [47], and Calleam Consulting [48]). In addition to that, Boehm and Basili [49] reported that the cost of fixing requirements-based problems increases rapidly the farther into the software development they are discovered.

Eliciting the suitable requirements produces benefits such as preventing errors, improving quality, and reducing risk in software projects (as shown in the studies of Procaccino et al. [15], and Jones [16]). As stated previously, requirements reuse has been proposed as an advanced requirements elicitation technique that can be a key asset in obtaining SRSs of better quality through more effective engineering.

Requirements reuse has been recently tackled from a number of different perspectives: publications in recent years of scientific works with requirements reuse proposals and studies associated to the topic in major journals and conference proceedings (e.g. Eriksson et al. [50], Hauksdottir et al. [51], Li et al. [52], Chernak [53], Carrillo-de-Gea et al. [54], Hoffmann et al. [55], and Pacheco et al. [56]); reports, as the one of Goldin and Berry [57], of experiences related to investigating the benefits of reuse in the requirements area; textbooks published very recently that include software reuse in their topics (as the one of Wiegers and Beatty [26]); organizations of workshops (fourth editions currently held of RePa@RE[1]) and tutorials (three tutorials on reuse through patterns in ICSE 2013-15[2]); etc.

On the other hand, there exist indicators that show an increase of interest in requirements reuse practices in industry: the latest releases of Requirements Management Tools (RMTs) are including requirements reuse as part of their functionality (e.g. IBM Rational Doors[3], Jama[4], and Visure Requirements[5]), and social media are witnessing the creation of groups dedicated to the topic (e.g. *Reuse in Requirements Engineering* LinkedIn group).

Requirement patterns represent one of the techniques to achieve requirements reuse. There are multiple proposals for reusing requirements through patterns (e.g. Withall [9], Kotula [37], and Keepence and Mannion [38]), but they differ in different aspects. Some of these aspects are: the artefacts to reuse, ranging from sentences in natural languages (e.g. Benitti and da Silva [58]) to more sophisticated elements (e.g. conceptual models in de Freitas et al. [59] and requirements formalization rules in Beckers and Heisel [60]); the

[1] International Workshop on Requirements Patterns http://www.utdallas.edu/~supakkul/repa15/
[2] International Conference on Software Engineering http://2015.icse-conferences.org/
[3] http://www-03.ibm.com/software/products/en/ratidoor/
[4] http://www.jamasoftware.com/
[5] http://www.visuresolutions.com/requirements-engineering-tool

reuse process proposed, with some proposals incorporating a detailed method to use the artefacts to reuse (e.g. van Lamsweerde [61]), others just presenting informal guidelines (e.g. Hoffman et al. [62]), and others not facing this aspect (e.g. Haeng-Kon [63]); and the availability of repositories of artefacts to reuse, with proposals that have such repository (e.g. Jaramillo [64]) and others that do not have it (e.g. López et al. [65]). Nevertheless, as it is presented in the State of the Art and the Practice (Chapter 2), there is not any widespread proposal used in organizations, nor any proposal that is complete enough in the sense of covering all the necessary elements to encourage organizations to adopt requirements reuse.

Therefore, the motivation behind this thesis is the need of a complete framework of requirements reuse through patterns that guides organizations during the reuse process in order to improve their RE processes.

1.3 Research Questions

This thesis proposes its own approach of Software Requirement Patterns (SRPs) and an entire framework around it to capitalise requirements knowledge that is convenient to reuse from previous IT projects. The framework includes all the necessary elements to encourage organizations to adopt it, and which will help to obtain SRSs of better quality with a more effective RE process. Therefore, the research objective is formulated in terms of the following design problem:

> *Define the concept of SRP and a framework around this concept in order to facilitate the management and update of catalogues of SRPs that encapsulate reusable requirements knowledge, and to ease their use during the requirements elicitation and specification of a project in order to obtain SRSs of higher quality with a more effective RE process.*

The main goal of the framework is the use of SRPs for reusing requirements knowledge (specifically, this knowledge corresponds to textual requirements) with the aim of improving the quality and validity of SRSs and reducing the time spent during requirements elicitation and specification processes, with the economic cost savings that both entail. The resulting framework is named PAttern-Based Requirements Elicitation (PABRE) framework.

The Research Questions (RQs) of this thesis are presented below, using the design science approach proposed by Wieringa [66] to classify them in Knowledge Problems (KP) and Practical Problems (PP):

RQ1 *(KP) In the field of requirements engineering, what are the pattern-based approaches proposed so far and what is the state of the practice of this kind of approaches?*

RQ2 *(PP) What is the best structure and semantics that SRPs should have to be applied over FRs, NFRs and NTRs?*

RQ3 *(PP) How can SRPs be integrated into the RE activities and processes so that their application yields benefits that justify the cost of their adoption?*

1.4 Methodological Approach

Shaw [67] states several ways of characterizing software engineering research, in terms of what she describes as research settings, research products, and validation techniques.

Research settings are the different classes of research problems. Shaw lists five research settings along with a sample question as example (Table 1.1). The settings of this thesis, in terms of Shaw's settings, are feasibility, characterization, and method/means. RQ1 tries to explore the *feasibility* of the existing approaches to patterns in the context of RE knowledge reuse; RQ2 is about the *characterization* of SRPs; RQ3 finds *means* to include SRPs in RE processes; finally, RQ3 attempts to *characterize* the potential benefits of using SRPs.

Research products are the tangible results of research. Shaw lists five research products along with a short description of how to achieve them (see Table 1.2). The research products of this thesis include a qualitative or descriptive model, a technique, a system, and an analytic model. *Descriptive models* corresponds to the study of the different approaches to patterns for reusing knowledge during RE, and to the empirical study carried out to understand the state of the practice of requirements reuse and requirement patterns. The *technique* produced includes the defined SRP structure, as well as the creation, use and update processes that aim to include SRPs during the RE stage. The *system* makes reference to the tools developed to manage and use SRPs according to the proposed process. Finally, the *analytic model* is a return-on-investment model on requirements reuse in general, and SRPs in particular.

Table 1.1 – Shaw's list of research settings

Research setting	Sample question
Feasibility	Is there an X, and what is it? Is it possible to accomplish X at all?
Characterization	What are the important characteristics of X? What is X like? What, exactly, do we mean by X? What are the varieties of X, and how are they related?
Method/Means	How can we accomplish X? What is a better way to accomplish X? How can I automate doing X?
Generalization	Is X always true of Y? Given X, what will Y be?
Selection	How do I decide between X and Y?

Table 1.2 – Shaw's list of research products

Research product	Research approach or method
Qualitative or descriptive model	Organize and report interesting observations about the world. Create and defend generalizations from real examples. Structure a problem area; formulate the right questions. Do a careful analysis of a system or its development.
Technique	Invent new ways to do some tasks, including procedures and implement-tation techniques. Develop a technique to choose among alternatives.
System	Embody result in a system, using the system development as both source of insight and carrier of results.
Empirical predictive model	Develop predictive models from observed data.
Analytic model	Develop structural (quantitative or symbolic) models that permit formal analysis.

Table 1.3 – Shaw's list of validation techniques

Technique	Character of validation
Persuasion	A technique, design or example.
Implementation	Of a system or technique.
Evaluation	With respect to a descriptive model, a qualitative model, an empirical quantitative model.
Analysis	Of an analytic formal model, an empirical predictive model.
Experience	Expressed in a qualitative or descriptive model, as decision criteria or an empirical predictive model.

For the research validation Shaw provides a list of five techniques (Table 1.3). The validation techniques used in this thesis are persuasion, implementation, evaluation, and experience. *Persuasion* is used all along this thesis using examples that illustrate the behaviour of the proposed ideas or processes. For the *implementation* technique, the implemented system shows the feasibility to use SRPs and the proposed methods to assist requirements engineers while eliciting and specifying requirements and managing SRPs. The *analysis* and academic validation of the PABRE framework is done with a retrospective analysis and an empirical study. The *evaluation* of the data gathered from the empirical studies is contrasted with the results obtained in similar studies. The design of the SRP structure, the methods, and the tool are based on the *experience* obtained from the SRSs provided by an organization and their requirements engineers.

1.5 Research Context and Contributions

The research in this thesis has been conducted within the Software and Service Engineering Research Group (GESSI[6]) of the Universitat Politècnica de Catalunya - BarcelonaTech (UPC). The GESSI group conducts research in many fields of software engineering, with particular emphasis on RE, software quality, software architecture, service-oriented computing, open source software, software modelling and empirical research.

This thesis is focused on the reuse of requirements knowledge, so it is directly connected to the RE research line. This line has been progressing through several projects and collaborations that GESSI carried and is currently carrying out. Some of the most representative are the research project Pros-Req: Requirement-Based Production of Service-Oriented Software [68] and the mobility project Requirements Engineering for Multi-stakeholder Distributed Systems [69].

As a result of one of the collaborations, GESSI worked with the Luxembourg Institute of Science and Technology (LIST[7]). This collaboration provided the context for defining a strategy that could contribute to improve the RE processes taking place at LIST. The preliminary proposal of this strategy was the basis for this thesis. It was presented in [70] [71] [72] and included: 1) A first definition of the structure of SRPs; 2) A first version of a set of Non-Functional SRPs; 3) A first version of a method to use SRPs during requirements elicitation; and 4) Some preliminary guidelines to update the set of SRPs. In 2009 this PhD candidate joined GESSI, and since then she has been working on updating this first version of SRPs, which was the initial point of this thesis, getting as a result the PABRE framework.

Having established the research questions in Section 1.3, the contributions of this thesis can be grouped into four areas: the use of patterns for reusing RE knowledge (RQ1), the definition of the metamodel of SRPs (RQ2), the integration of SRPs in RE processes (RQ3), and the empirical research about SRPs (which validates the results of RQ2 and RQ3).

[6] http://www.essi.upc.edu/~gessi/
[7] http://www.list.lu/

Some of the contributions of this thesis have been published in venues such as the IEEE International Requirements Engineering Conference (RE), the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), and the Symposium on Applied Computing (SAC). In addition, there is a book chapter on this thesis topic. Also, some works have been published in workshops such as the International Workshop on Requirement Patterns (RePa) held at the RE conference, and the Empirical Fair at REFSQ. Likewise, the applicant presented the work in the Doctoral Symposium at REFSQ [73], getting good feedback from the chairs. It is also worth mentioning the applicant's Bachelor and Master Thesis, along with the PhD Thesis Proposal, since the three of them represent important milestones for the research done in this thesis. Finally, there is a paper submitted (but still on review process) to the Empirical Software Engineering Journal (EMSE), and two others on the way targeted to be published at Information and Software Technology Journal (IST) and the Requirements Engineering Journal (REJ).

In the following, the contributions for each one of the four areas before-mentioned are presented.

1.5.1 Use of Patterns for Reusing Knowledge during RE

The main contribution in this area is twofold. Firstly, a Systematic Mapping (SMAP) was done, following the guidelines provided by Petersen et al. [74] and Kitchenham and Charters [75]. The focus of the SMAP are publications related to RE that deal with the use of patterns for reusing knowledge during the elicitation and specification of requirements. The SMAP is focused on patterns for these two RE activities since this thesis deals with both of them. There are no published works in relation to this study, although the applicant is working on a paper about the topic, which is targeted to be published at IST. Currently, the results of the SMAP can be found in Section 2.2 of this thesis. Previous versions of it were presented in the applicant's Master Thesis [76] and the PhD Thesis Proposal [77].

Secondly, an empirical study was carried out with the aim of exploring the current and potential use of requirement patterns-based approaches in industry. To do so, an exploratory survey addressed to IT people with industrial experience in RE was conducted. The design of the survey was first proposed in the Empirical Fair at the

REFSQ conference in 2013 [78]. The first results of the survey were published at the REFSQ conference in 2014 [79]. A complete analysis of the results was submitted in 2015 to the Empirical Software Engineering journal (EMSE), but it is still on review process [80].

1.5.2 Definition of the SRP Structure

The contribution in this area is the basis of the PABRE framework, which is the metamodel of SRPs. A preliminary proposal of the SRP structure existed before the applicant started her work with GESSI. Afterwards, during the thesis, the structure was extended and rigorously modelled as a metamodel, which is presented as a UML diagram. This work was done through the observation and analysis of SRSs, the study of related work that supports the proposed structure, and the observations and judgement provided by RE experts. In order to achieve the goal of providing a complete framework for requirements reuse through SRPs, the metamodel also includes the structure proposed for the catalogue or repository of SRPs.

The first version of the metamodel was derived from the observation of the NFRs in SRS documents and was presented in the REFSQ conference in 2010 [81] and later on as part of the Master Thesis in 2011 [76]. Next, the metamodel was validated and checked regarding the observation of NTRs and FRs. An evolved version of the metamodel was presented in the Thesis Proposal at 2013 [77] and the chapter in the book Managing Requirements Knowledge in 2013 [82]. The final proposal of the metamodel has not been published yet, but the applicant is currently working on a publication that incorporates it, which will be submitted to REJ. In the meantime, the last version of the metamodel can be found in Chapter 4 of this thesis.

During the proposal of the metamodel, a catalogue of SRPs corresponding to the analysed SRSs was constructed. The first version of Non-Functional SRPs (NF-SRPs) was proposed before the applicant joined GESSI and accepted to carry out the research resulting in this thesis. Afterwards, the applicant adapted the existing NF-SRPs to the definitive metamodel. The Non-Technical SRPs (NT-SRPs) were included as result of the Master Thesis in 2011 [76] and presented as a publication in the RePa@RE workshop in 2012 [83]. The first version of the Functional SRPs (F-SRPs) was published at the SAC conference in 2013 [84]. Evolved versions of the NF-SRP, NT-SRP and F-SRP catalogues can be found in the Thesis Proposal [77]. The NT-SRP catalogue was also published as part of the book chapter [82]. In order to homogenise the three catalogues and improve their quality, a final version of them was constructed, which correspond to the versions presented in this thesis.

1.5.3 Integration of SRPs in RE processes

The PABRE framework is complemented by a set of assets to guide the management and use of an SRP catalogue. These assets correspond to four contributions:

- The proposal of different approaches for the use of SRP catalogues during requirements elicitation and specification processes. A preliminary proposal of a guided method to use SRPs existed before the applicant started her work with GESSI. Later on, it was improved, presenting it with dissemination purposes in the chapter inside the book Managing Requirements Knowledge in 2013 [82] and in the Requirements Engineering Magazine in 2014 [85]. The last version of this method, as well as some other less-heavy approaches for using SRPs, are presented in this thesis.
- The definition of methods to manage an SRP catalogue (i.e. for constructing the catalogue and updating it). The method for constructing SRPs was first presented as part of the Master Thesis in 2011 [76], and later on improved and published as part of the book chapter in 2013 [82]. Before the applicant started her work with GESSI, there existed preliminary guides to update an SRP catalogue. In the same book chapter stated before, a complete guide the evolution of an SRP catalogue and maintaining it up-to-date was presented. The last version of both methods are presented in this thesis.
- The design of an economic model to perform cost-benefit analysis on the adoption of SRPs based on Return-On-Investment (ROI). A first version of this ROI model was included in the Thesis Proposal presented in 2013 [77]. Its last version is presented in this thesis.
- The design and implementation of the PABRE system as technological support to the management and use of SRPs. This system is composed by PABRE-Man, which is a subsystem to manage an SRP catalogue (mainly developed as result of the applicant's Bachelor Thesis in 2010 [86]), PABRE-Proj, which is a subsystem to use

an SRP catalogue during elicitation and specification processes (developed thanks to the applicant's Bachelor Thesis and other Bachelor and Master Thesis supervised by the applicant and one of her advisors [87] [88] [89] [90] [91]), and PABRE-WS, which is a subsystem that provides REST web services open to any RMT for including SRPs in their functionalities (developed thanks to a Master Thesis [92]). In all the cases, the development and implementation was guided by the applicant. The PABRE-Man subsystem was presented as a Demo at the RE conference in 2011 (achieving the best poster award) [93]. Likewise, the PABRE-Proj subsystem was presented as a Demo in the 2013 edition of the RE conference [94]. In addition, a general overview of the PABRE system was published with dissemination purposes in the Requirements Engineering Magazine in 2014 [85].

1.5.4 Empirical Research about Requirement Reuse and Requirement Patterns

The PABRE framework was advertised in the Empirical Fair at the REFSQ conference in 2011 [95]. The goal was to attract organizations that were interested in trying the SRPs proposed inside the framework with the aim of conducting some type of empirical study. Although no collaboration came up from this event, some comments were done that were useful to improve the framework.

In addition, an empirical study was produced as result of a research stay of the applicant at the Blekinge Technical University. In collaboration with Professor Tony Gorschek, the acceptance of the overall PABRE framework proposal, as well as the results obtained in the survey-based empirical study, were validated by interviewing 22 RE professionals of 11 Swedish organizations. At the moment of writing this thesis, the results of this study have not been published yet.

1.6 List of Publications

The list of accepted publications related to the PhD thesis are shown in Table 1.4, while the list of publications in progress (i.e. not accepted yet) are shown in Table 1.5. In both tables the publications are organized by their type, and they contain for each publication the research questions they contribute to.

Table 1.4 – List of accepted publications

Ref.	Venue	Title	Year	Remarks[a]	RQs
Book Chapters					
[82]	Managing Requirements Knowledge	Constructing and Using Software Requirements Patterns	2013	Published in Springer	RQ2, RQ3
Journals not indexed in the JCR					
[85]	Requirements Engineering Magazine	Requirements Reuse with the PABRE Framework	2014		RQ3
International Conference Proceedings					
[81]	REFSQ	A Metamodel for Software Requirement Patterns	2010	CORE-B, Short paper	RQ2
[84]	SAC	A Catalogue of Functional Software Requirement Patterns for the Domain of Content Management Systems	2013	CORE-B, Regular paper	RQ2
[79]	REFSQ	Requirements reuse and patterns: A Survey	2014	CORE-B, Short paper	RQ1
Demos at International Conferences					
[93]	RE	PABRE-Man: Management of a Requirement Patterns Catalogue	2011	CORE-A, Best poster award	RQ3
[94]	RE	PABRE-Proj: Applying Patterns in Requirements Elicitation	2013	CORE-A	RQ3
Workshop Proceedings					
[95]	Empirical Fair Track @ REFSQ	Interested in Improving Your Requirements Engineering Process? Try Requirement Patterns!	2011	CORE-B[b]	RQ3
[83]	RePa @ RE	A Catalogue of Non-Technical Requirement Patterns	2012	CORE-A[b]	RQ2
[78]	Empirical Fair Track @ REFSQ	Online Questionnaire: Using a Pattern Catalogue in Requirements Engineering Activities	2013	CORE-B[b]	RQ1
Doctoral Symposiums					
[73]	REFSQ	Definition and Use of Software Requirement Patterns in Requirements Engineering Activities	2014	CORE-B[b]	---
Technical Reports					
[86]	Bachelor Thesis @ FIB-UPC	Implementació del procés d'utilització de patrons de requisits[c]	2010		RQ3
[76]	Master Thesis @ LSI-UPC	Including Functional and Non-Technical Requirements in a Software Requirement Patterns Catalogue	2011		RQ1, RQ2, RQ3
[77]	Thesis Proposal @ LSI-UPC	Definition and Use of Software Requirement Patterns in Requirements Engineering Activities	2013		RQ1, RQ2, RQ3

[a] CORE classifications correspond to 2014
[b] CORE are the ones of the main conference in which the workshop or symposium was held
[c] English translation: Implementation of the Process for Using Requirement Patterns

Table 1.5 – List of publications in progress

Ref.	Venue	Title	Year	Remarks[a]	RQs
Journals indexed in the JCR					
[80]	EMSE	Requirements Reuse and Requirement Patterns: A State of the Practice Survey	2015	submitted, IF=2.161	RQ1
n/a	IST	Requirement Patterns for Elication and Specification: A Systematic Mapping	2016	no submitted, IF= 1.046	RQ1
n/a	REJ	A Software Requirement Patterns Metamodel	2016	no submitted, IF= 0.882	RQ2

[a] Impact factors correspond to 2014

1.7 Structure of this Thesis

This thesis is structured into two parts. Part I covers the state of the art, and the definition and implementation of the PABRE framework, which are related to RQ1, RQ2, and RQ3. Part II covers the validation conducted on the PABRE framework, which corresponds to validating the results of RQ2 and RQ3. Table 1.6 contains the relationship between the chapters, the research questions and the publications used as main sources for the chapters' contents.

Table 1.6 – Summary of chapters of this dissertation

Part	Chapter	Title	RQ	Source of contributions
	1	Introduction	---	---
I	2	Requirements Reuse and Patterns: State of the Art and the Practice	RQ1	[76] [77] [78] [79]
I	3	Overview of the PABRE framework	RQ2, RQ3	---
I	4	Metamodel of an SRP catalogue	RQ2	[76] [77] [81] [82]
I	5	Use of an SRP catalogue	RQ3	[82] [85]
I	6	Lifecycle of an SRP catalogue	RQ3	[76] [85]
I	7	Catalogue of SRPs	RQ2	[76] [77] [82] [83] [84]
I	8	Return-on-Investment of an SRP catalogue	RQ3	[77]
I	9	The PABRE system	RQ3	[85] [86] [93] [94]
II	10	Validation of the SRP metamodel	RQ2	---
II	11	Validation of the PABRE framework	RQ3	[80] [95]
	12	Conclusions	---	---

Part I. Chapter 2 describes a systematic mapping conducted to study the different approaches that propose the use of patterns as means to capitalize the reuse of knowledge during RE as well as an empirical study conducted in order to explore the current state of the practice of requirement patterns-based approaches. Chapter 3 presents the PABRE framework in a general way and explains what is understood by SRP in PABRE. Chapter 4 provides the definition of the SRP metamodel. Chapter 5 and 6 go deeper on the use of SRPs, and present the methods to use SRPs during elicitation and specification processes, and the methods to create SRPs and update them, respectively. Chapter 7 presents the catalogues of SRPs for NFRs, NTRs and FRs. Chapter 8 describes the Return-on-Investment model on the adoption of SRPs. Finally, Chapter 9 contains the implementation of the proposed framework.

Part II. Chapter 10 presents the validation done over the SRP metamodel to check that it is useful for the three types of requirements (NFRs, NTRs and FRs). Chapter 11 explains an empirical study conducted in order to explore the perception of IT practitioners about the general viability of SRPs.

Finally, Chapter 12 provides the conclusions of this thesis as well as the future work.

Part I

The PABRE Framework

CHAPTER 2 Requirements Reuse and Patterns: State of the Art and the Practice

2.1 Introduction

Software engineering has been significantly impacted since the concept of patterns was adopted by researchers and practitioners. As explained by Alexander [96], when a problem occurs over and over in an environment, its generalized solution along with certain forces that govern the application of this solution is coupled to form a pattern. One can reuse pattern's knowledge to solve recurring problems, without ever doing it in the same way twice.

As in any other software engineering discipline, patterns have been a matter of research in RE. This is why the research question RQ1 (Table 2.1) aims at investigating the current state of the requirement pattern-based approaches. However, the current state can be explored from either a state of the art perspective (a more academic point of view) or a state of the practice perspective (a more practical point of view). As a consequence, RQ1 has been decomposed in subquestions RQ 1.1 and RQ 1.2, which address each one of these perspectives, respectively (Table 2.1).

Table 2.1 – Research question 1 and its subquestions

RQ 1	**In the field of requirements engineering, what are the pattern-based approaches proposed so far and what is the state of the practice of this kind of approaches?**
RQ 1.1	In the field of requirements engineering, do pattern-based approaches proposed so far provide a ready-to-use technique to achieve RE knowledge reuse during the elicitation and specification of requirements?
RQ 1.2	In the field of requirements engineering, what is the state of the practice of requirements reuse techniques in industry, and specifically of the techniques based on patterns?

In the following, the studies related to research questions RQ 1.1 and RQ 1.2 are presented. The first study corresponds to an SMAP focused on patterns for reusing RE knowledge during the elicitation and specification of requirements (Section 2.2). The second study corresponds to an empirical study carried out with the aim of exploring the current and potential use of requirement patterns-based approaches in industry (Section 2.3). Finally, a general conclusion for both studies is given (Section 2.4).

2.2 State of the Art: Patterns for Reuse on RE

As patterns have been a matter of research in RE, all kind of proposals and approaches for eliciting and specifying requirements using patterns with RE knowledge have emerged. RE knowledge embraces requirements knowledge (i.e. knowledge directly related to the requirements per se, such as requirements sentences, the structure or writing of requirements, and models related to requirements) as well as other knowledge related to RE (such as RE best practices). Maybe the first question that may be raised related to the matter is: how and what knowledge do patterns capture to help users overcome inherent RE problems? The answer to this question requires determining what proposals exists, understanding them and assessing all of them together under the same criteria and a common framework. The proposals may diverge in several matters: the structure of capitalized knowledge, the language in which they are expressed, the classification and browsing capabilities of the repository, the existence of a method for building, updating, and exploiting the RE knowledge repository, etc. Therefore, it is important to conclude too which is the most consolidated body of knowledge and on the contrary, which are the most remarkable gaps to bridge.

The aim of this section is to identify and evaluate the different approaches of patterns for reusing RE knowledge during the elicitation and specification activities in literature. The review is focused on patterns for these RE activities since this thesis deals with both of them. A systematic mapping was conducted to accurately retrieve and analyse the proposals, defining and conducting a rigorous protocol following the guidelines described by Petersen et al. [74] and Kitchenham and Charters [75]. As a result of such research, it has been possible to evaluate the current state of the art in RE patterns for eliciting and specifying requirements and to identify the strengths and weaknesses of its current status.

2.2.1 Systematic Mapping Studies

Systematic Mapping Studies (SMAPs) or scoping studies are designed to give an overview of a research area through classification and to count contributions in relation to the categories proposed in that classification [75] [97]. They involve searching the literature in order to know what topics have been covered, and where they have been published [97]. Although an SMAP and a Systematic Literature Review (SLR) share some commonalities (e.g. with respect to searching and study selection), they are different in terms of goals and, thus, in the approaches to data analysis. While SLRs aim at synthesizing evidence, also considering the strength of evidence, SMAPs are primarily concerned with structuring a research area.

To conduct any type of literature study in an accurate and objective manner, it is necessary to use a precise and rigorous methodology. For such a purpose, the principles and guidelines for performing an SMAP defined by Petersen et al. [74] have been followed in this thesis. Whenever Petersen's et al. guidelines were not specific enough, SLR guidelines defined by Kitchenham and Charters [75] were consulted too. Both guidelines have been derived from other existing studies used by medical researchers and adapted to reflect the specific problems of software engineering research. When applied properly, they drastically reduce the risk of bias and incompleteness in the review results.

The stages of the methodology, as defined by Petersen et al. [74], are as follows:

- **Planning the mapping.** Activities performed before conducting the SMAP. They include the identification of the need of the study, its scoping, and the definition of the protocol that specifies the criteria that will be used to perform the review (e.g. search keywords, bibliographic sources, selection criteria, etc.).
- **Conducting the mapping.** Activities that constitute the execution of the SMAP, following the protocol previously defined. They include the identification of studies, the selection of primary studies, and the data extraction and classification.
- **Reporting the mapping.** Activities to report the results of the SMAP. It includes the specification of a dissemination mechanism, the format of the report, and its evaluation.

2.2.2 Planning the SMAP

During the planning, all the decisions relevant to the conduction of the SMAP are made. Concretely, it is necessary to identify the need for the study, to scope it by defining the research questions the study is willing to answer, and to define the protocol that will be followed during the study. Each one of these three points is further explained in the following subsections.

2.2.2.1 Identification of the Need for a Study

As Petersen et al. [74] states, prior to undertaking an SMAP, it is necessary to ensure that such a study is required by searching for others in the subject and assessing their quality through a quality assessment checklist.

There is no procedure defined in Petersen's et al. [74] and Kitchenham and Charters' [75] guidelines in order to implement this search. However, also to make this step systematic, two tactics were applied. First, to increase the number of results, not only other SMAPs were searched, but all type of reviews and state of the art documents, despite of the methodology followed for developing them. Second, a procedure analogous to the main search of the systematic mapping presented in this section was followed. That is, a search protocol to identify other reviews was defined. Such protocol was based on the protocol defined in the main search, which will be explained in the following subsections. Hence, other reviews were searched for once the systematic mapping protocol was defined and before the systematic mapping was conducted. In a glance, automatic searches were done over the same databases and resources, using the same keywords with the addition of the following terms: "state of the art", "SLR", "survey", "review"[8] and "systematic mapping". As a result, 408 papers were found fulfilling the search criteria. From them, 15 papers were selected by title (the list is available in Appendix I). There were some existing systematic reviews on requirements reuse in general (e.g. de Azambuja et al. [98]) or focused on the security domain (Mellado et al. [99] and Souag et al. [100]). However, none of them presented a review on how patterns are used for requirements reuse during requirements elicitation and specification.

Therefore, after performing the searches, no literature study was found on applying patterns for reusing RE knowledge at the elicitation and specification activities.

2.2.2.2 Scoping the Study

The next step of the review is the formulation of the research question, which in turn, will drive the review methodology. To formulate it, the PICO structure, summarised in Kitchenham and Charters' guidelines [75], was followed. PICO is an acronym that stands

[8] This term was used instead of the more general term "systematic literature review", to make the search more robust with respect to terminological variations.

for Population, Intervention, Comparison and Outcome. It consists of explicitly identifying these concepts in the research question in order to derive later the keywords to perform the search. In the study presented in this section, though, the Comparison is more a kind of general analysis of the field, since it is not aimed at ranking the proposals found or to compare to some other existing approaches. The research question that drove the systematic mapping, which has been presented in Section 2.1, was decomposed in four subresearch questions (Table 2.2). In the same table, the Population (P), Intervention (I) and Outcome (O) are identified from the main research question.

Table 2.2 – Research question 1.1 and its subquestions

RQ 1.1	In the field of *requirements engineering* (P), do *pattern-based approaches* (I) proposed so far provide *a ready-to-use technique to achieve RE knowledge reuse during the elicitation and specification of requirements* (O)?
RQ 1.1.1	What is the chronological overview of the research done so far in patterns for reusing RE knowledge?
RQ 1.1.2	What are the characteristics of the proposed patterns?
RQ 1.1.3	Which assets are provided with the proposed patterns?
RQ 1.1.4	What are the most consolidated patterns?

2.2.2.3 Defining the Protocol of the Study

The protocol followed during the study defines three important aspects related to the finding of primary studies: the bibliographic sources where the searches to look for studies will be done, the keywords that will be used for doing such searches, and finally the inclusion and exclusion criteria used for selecting the primary studies over all the set of found studies.

Bibliographic sources. The search process can be either automatic through the usage of bibliographic databases or manual through gathering the works from specific known journals and conferences of the target field. Kitchenham et al. [101] analyses the drawbacks of both approaches through a case study. Their work confirms that broad automated searches find more relevant studies than manual searches. Because of that, it was decided to follow in this study the automatic search approach. The selected databases were: ACM Digital Library, IEEE Xplore, Science Direct and Springer Link. These databases cover the most important conferences and journals related to RE (such as the Requirements Engineering Journal, the IEEE Requirements Engineering Conference, and the International Working Conference on Requirements: Foundation for Software Quality), as well as some other important ones related to software engineering in general (such as the IEEE Transactions on Software Engineering, the Information Systems Journal, and the Empirical Software Engineering Journal).

Keywords. The keywords used for the search were retrieved from the PICO terms of the research question (Table 2.3). Particularly, they were extracted from the Population and Intervention. In Kitchenham and Charters' guidelines [75], it is also suggested to extract the keywords from the Comparison and Outcome, which is the common procedure in the field of medicine. However, as it is also stated by Kitchenham [102] and identified by Petersen et al. [97] and other SLRs [103], this is not always applicable. For instance, as stated by Kitchenham and Charters [75], it is applicable in SLRs when performing a comparison between two already known different approaches. However, this study was evaluating the different proposed approaches found in the literature. Similarly, the outcome in the research question was not based on a particular measurement and hence it could not be included as a keyword.

Table 2.3 – Keywords related to the Population and Intervention

PICO	Abstraction	Terms
Population: requirements engineering	requirements	requirement, requirements
Intervention: pattern	reusable	reuse, reused, reusable

In order to get as much work as possible related to RE patterns (which correspond to the combination of the Population and Intervention terms in the research question), both terms were generalized: firstly, 'requirements engineering' was abstracted to 'requirements', as works known before the study never talked about requirements reuse but about requirements engineering reuse; secondly, 'patterns' was abstracted to 'reusable', as there was no an strict use of the term pattern in the literature when works were actually dealing with some kind of patterns for reusing RE knowledge (some works known before the study talked about templates [104], boilerplates [105] or just reusable structures in general [106], which at the end are kind of patterns). Later on, for the abstractions of the terms of the Population and Intervention of the research questions, a set of variants were identified (Table 2.3).

The resulting query was obtained by combining the Population and Intervention terms, which using wildcards were simplified to (requirement* AND reuse*). The first aim was to use the search string just to the title, abstract and keywords of the papers. However, when reviewing the databases, it was noticed that one of the chosen databases (Springer Link) did not allow to search just in these specific fields. For this database, the search was done over the full-text of the article, filtering afterwards the results for those categories that could include studies related to software engineering (Software Engineering, Information Systems and Applications, and Business Information Systems).

Inclusion and exclusion criteria. The following inclusion criteria were applied to titles, abstracts and full-text reading of papers while screening them:

- I1: Papers published until December 2014 (included) and accessible through the chosen bibliographical sources.
- I2: As the writing of the first version of this thesis was finished in late 2015, papers published in the most important RE journals and conferences during 2015 were manually included too, those ones being: Requirements Engineering Journal, the IEEE Requirements Engineering Conference, and the International Working Conference on Requirements: Foundation for Software Quality.
- I3: Papers related to the topic of this study: proposals of patterns for knowledge reuse during RE. As mentioned before, the search included papers that, although they could be considered patterns for reusing RE knowledge, the authors actually did not mention the term pattern in the work.
  - I3.1: Papers present a RE reuse strategy as a contribution of the paper.
  - I3.2: Papers define explicitly the RE reuse strategy.
  - I3.3: Papers present a reuse strategy based on patterns.
  - I3.4: Papers present a reuse strategy for the elicitation and/or specification of requirements.
  - I3.5: Papers present the structure of patterns in which the reuse strategy is based.

The following criteria stated when a study was excluded:

- E1: Papers present summaries of conferences/editorials.
- E2: Papers not accessible in full-text.
- E3: Papers not available in electronic form.
- E4: Papers no written in English.
- E5: Papers present SLRs or SMAPs (in the case of SLRs and SMAPs, their references were always checked to include any paper that might be missing in the automatic searches).
- E6: Papers present non-peer reviewed material (except those ones that came from the same authors of a peer-reviewed material and were necessary for fully

understanding the presented approach). The only exception for this rule was Withall's book "Software Requirement Patterns" [SMAP-101], corresponding to proposal P77, since the topic of the book perfectly matched the SMAP topic.

2.2.3 Conducting the SMAP

The SMAP was conducted by following the protocol defined in the previous subsection. The activities of this phase included: the identification of studies by selecting papers using inclusion and exclusion criteria, and the data extraction from the studies and their later classification.

2.2.3.1 Selection of Primary Studies

The search was conducted in ACM Digital Library, IEEE Xplore, Science Direct and Springer Link (inclusion criterion I1). For the conference and journals editions in year 2015, the same search was conducted manually over title, abstract and keywords (inclusion criterion I2).

After retrieving the results, several steps were conducted to filter the candidates.

Deletion of duplicates and non-relevant results. Results that were duplicated were removed and also those ones that were not relevant according to the exclusion criterion E1 (for instance those results that were the introduction of proceedings).

Selection by title. The objective of this first filter was to quickly identify and remove noise from the results. After this selection, documents whose scope was clearly unrelated to RE were removed (inclusion criterion I3.1).

Selection by abstract. At this stage, all those works that although being related to RE, the reuse of knowledge was not their main contribution, were discarded (inclusion criterion I3.1).

Selection by full paper (fast reading). Each work was skimmed to be sure before the next stage that it was relevant for the systematic review. Firstly, according the exclusion criteria E2, E3 and E4, the papers that were not accessible in full-text (asking for it to the authors when necessary), not accessible in electronic format, or that were not written in English were removed. Secondly, the papers that were SLRs or SMAPs were discarded (exclusion criterion E5); in that case, tough, the references were checked for possible addition of further work. Finally, from the resulting set, only the papers that accomplished properly the inclusion criteria I3.1, I3.2, I3.3 and I3.4 were selected, i.e.: (1) presented a RE reuse strategy as one of the contributions of the paper; (2) defined explicitly the reuse strategy; (3) the reuse strategy was based on patterns; (4) the reuse strategy was used for requirements elicitation and/or specification.

Selection by full paper (detailed reading). The deep reading of the article was made by marking the relevant parts and annotating comments to ensure that future readings will take less time. During this step, a template related to keywords and other important issues was filled in order to classify and facilitate the future data extraction and synthesis of those works that were selected in this step. At this point, papers presenting the structure of the patterns in which the RE reuse strategy was based on (inclusion criterion I3.5) were the ones selected as the primary studies for this SMAP.

Addition of further work (snowballing). During the SMAP process, other works were included through the process of backward snowballing (both from selected works or other SLRs or SMAPs). In this sense, works included were: the ones that were the basis for the development of the reuse strategy in the retrieved papers (when necessary), or relevant citations to this SMAP topic not found in the automatic/manual searches. The works satisfying the same inclusion and exclusion criteria as previously, and not discarded by the exclusion criterion E6 (i.e. the work was peer-reviewed), were added into the SMAP.

From the searches, 755 papers were automatically retrieved from ACM Digital Library, 3688 from IEEE Xplore, 450 from Science Direct and 2884 from Springer Link. Then, 6 papers were added from the manual searches from the selected journals and conferences corresponding to 2015, leading to 7783 papers found. From the initial set of 7783 papers, 232 were removed as they were repeated in the selected databases, leading to 7551 papers. Afterwards, 6896 were discarded by title and 308 papers were excluded by abstract, resulting in 347 papers to evaluate by full paper. The works were evaluated first by a fast reading of the full paper, where 232 papers were discarded, and later on by a detailed reading, where 49 papers were removed, resulting in 66 papers to include in the systematic mapping. Finally, 38 additional papers were added through snowballing resulting in the final 104 papers. A summary of the selection process is depicted in Figure 2.1.

Figure 2.1 – Selection process of the systematic mapping

2.2.3.2 Data Extraction

To extract data from the identified primary studies, the template shown in Table 2.4 was developed. Each data extraction field had a data item and its description. The *General Aspects* category refers to metadata of the article. The *Overview of the Proposal* deals with general aspects of what is being reused, and the *Structure of the Proposal* with the structure in which the artefact to reuse is surrounded. The *Organization of the Proposal* is associated to the possible relationships among the artefacts to be reused and their classification, and the *Metamodel of the Proposal* to the possible metamodels that may exists for the artefacts to be reused or their classifications. The *Methods of the Proposal* talks about the existence of methods to manage the artefacts to reuse, and the *Catalogue of the Proposal* about the existence of set of artefacts ready to be reused. Finally, the *Tools and Validation of the Proposals* category makes reference to the existence of tools to facilitate the management of the artefacts to reuse and to the testing of the proposal.

Table 2.4 – Data extraction form

Data item		Description
General Aspects	**Study ID**	ID assigned
	Article title	Name of the article
	Author(s) name(s)	Set of name of the authors
	Author(s) countries	Set of countries of the authors
	Year of publication	Calendar year
	Venue	Name of publication venue
Proposal – Overview	**General classification of the proposal**	Classification related to the type of pattern proposed
	Scope of the proposal	Elicitation and/or Specification
	Domain of the proposal	Can the proposal be used in all domains or it is for a specific software domain? Which domain?
	Type of requirements to reuse	Functional requirements, Non-functional requirements, Non-technical requirements, some of them
	Artefact to reuse	Sentences in natural language, Conceptual models, Use cases, etc.
	Notation used to define the artefact to reuse	Natural language, Modelling language (e.g. UML, *i**, ad-hoc), Formal language (e.g. Description logics, Formal temporal logic), etc.

Proposal – Structure	**Structure in which the artefact to reuse is contained**	Has the artefact to reuse complementary information, apart from the artefact itself, as for instance the context of use, the problem that it solves, information relevant for further development stages, etc.?
	Structure information	What is the specific information contained with the artefact to reuse?
Proposal – Organization	**Relationships**	Is there any type of relationships considered among the artefacts to reuse? What are they?
	Arrangement	Are the artefacts to reuse arranged in a taxonomy or classification schema?
	Classification	Is there a specific classification proposed for the artefacts to reuse?
Proposal - Metamodel	**Reuse artefact metamodel**	Is there a metamodel to describe the artefacts to reuse?
	Arrangement metamodel	Is there a metamodel to describe the classification of the artefacts to reuse (if such classification exists)?
Proposal - Methods	**Artefact to reuse construction method**	Is there a method or guidelines to construct and define the artefacts to reuse?
	Reuse method	Is there a method or guidelines to reuse the artefacts?
	Update method	Is there a method or guidelines to update the artefacts?
Proposal - Catalogue	**Catalogue**	Is there an established set of artefacts to reuse?
	Catalogue domain	Is it general or specific for a project or test?
	Catalogue state	Is it a finished set or in evolution?
Proposal – Tools & Validation	**Related tools**	Does the proposal have related tools? What are the functionalities covered by them?
	Tests	Has the proposal been tested in real cases?
	Type of research	Empirical, Non-Empirical, etc.

2.2.3.3 Analysis and Classification Process

The information for each data item was graphically represented or tabulated (see Subsection 2.2.4). When more than one primary study was found presenting the same approach, they were considered as being part of the same proposal. All the primary studies identified for a proposal were studied, so proposals corresponded to a compendium of all the information found about it. The analysis was done over proposals. However, for the *General Aspects* category (Table 2.4), the data gathered was not possible to be analysed grouping by proposals (e.g. for the venue of publication, usually a proposal included material published in different venues). In that case, the analysis was done directly over the publications.

On the one hand, for each data item, the researchers of this study (the applicant and one of her advisors) grouped the proposals (or primary studies) and gave them a theme during analysis, which were discovered as the analysis progressed. Thereafter, the proposals belonging to each theme were counted. As an example, for the general classification of the proposal six main themes were identified: Templates, Patterns with context information, Domain/Non-Functional (NF) patterns, Formalization patterns, Writing guidelines patterns, and Process patterns. In addition, for some of the data items cross-tabulation analysis with other data items was performed.

On the other hand, it was also interesting to know what proposals were the most consolidated ones. For that end, the proposals were individually analysed. Specifically, it was interesting to know what proposals were dealing with:

- Both the elicitation and specification of requirements, so the proposals were as broad as possible in terms of RE stages.
- A general domain, so the proposals could be extrapolated to all domains.
- All types of requirements, so the proposals could be applied for all types, not being necessary to apply different proposals for different requirements types.
- Metamodels for the artefacts to reuse and their classification, so the proposals underwent through a deep analysis, construction and development.
- Methods for constructing, using and updating the artefacts to reuse, so the proposals took into account the different lifecycle stages of the artefacts to reuse.
- A ready-to-use set of classified artefacts to reuse, so the proposals incorporated complete examples that served as a guide for creating future artefacts to reuse.
- Tool(s) to facilitate the use and management of the artefacts to reuse, so the proposals were easier to be integrated in real environments.
- Validation in real scenarios, so the proposals provided evidence that they worked in real sets.

2.2.4 Reporting the SMAP

This subsection covers the reporting of the SMAP, that being split into presenting the results (Subsection 2.2.4.1) and discussing the threats to validity (Subsection 2.2.4.2).

2.2.4.1 Results

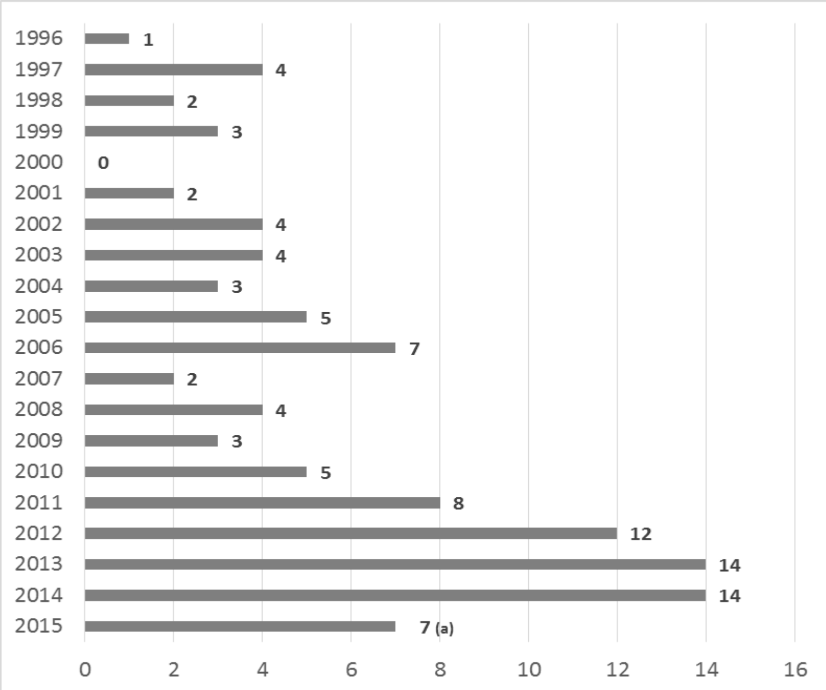
The SMAP was conducted according to the criteria and protocols defined in the previous subsections. First, the 104 selected works were read individually and the information therein was consolidated. The 104 works presented 80 different proposals of requirements reuse knowledge by using some kind of patterns (i.e. some proposals are explained in more than one work). The proposals were summarized in tables that are included at the end of this document in Appendix II; each table includes the data extraction form filled for each one of the proposals.

Next, the analysis of each one of the data items groups identified in the data extraction form are presented. All the analysis are done at the level of proposal, except for the *General Aspects* group, since it does not make sense to analyse the metadata of the articles at the level of proposal.

General Aspects

Year of publication. Figure 2.2 shows the number of papers identified according to their year of publication. The first work was published by Darimont et al. [SMAP-14], and it was the only one in 1996. While the interest in requirements reuse knowledge by using patterns was moderately stable in the 1997–2010, a significant increase can be observed since 2011. This increase in the number of works published indicates that patterns for requirements reuse are being considered more relevant by the software engineering research community. Besides an increased interest, another potential reason may be an increase in pressure to achieve information systems in a lower time because of the technology explosion of the last years.

Venues of publication. In this study, peer-reviewed venues (including journals, as well as peer-reviewed conferences and workshops) were considered. When necessary, non-peer-reviewed material was also included if it helped to understand a proposal presented in a peer-reviewed paper. In addition, the non-peer reviewed book "Software requirement patterns" [SMAP-101] was also considered, since its topic matches perfectly the topic of this SMAP. Figure 2.3 provides an overview of the distribution of the primary studies between these venues; the non-peer reviewed material corresponds to *books* and *internal reports*. A similar number of articles is published in international workshops and scientific journals, while there is a higher number of contributions published in international conferences. As expected, books and books chapters are the less common type of publication.



[a] The analysis for the primary studies in 2015 was done manually (see Subsection 2.2.2.3 for further information)

Figure 2.2 – Primary studies per year

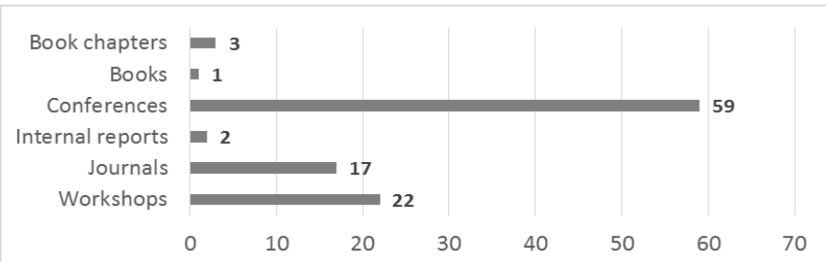


Figure 2.3 – Primary studies per venue types

Table 2.5 – Most targeted venues by the primary studies

Rank	Venue	Number of primary studies
1	**RePa** - IEEE International Workshop on Requirement Patterns	13
2	**RE** - IEEE International Requirements Engineering Conference	10
3	**REFSQ** – International Working Conference on Requirements Engineering: Foundations for Software Quality	4
4	**ICSE** - IEEE International Conference on Software Engineering	4
5	**RCIS** - International Conference on Research Challenges in Information Systems	3

Looking at the specific venues, it is also clear which ones are the most targeted by authors of requirements reuse approaches proposing patterns. The top five venues are stated in Table 2.5; the rest of venues only have one or two primary studies published on them. 13 articles have been published in the IEEE International Workshop on Requirement Patterns (RePa), which is a workshop held as part of the IEEE International Requirements Engineering Conference (RE). This conference is, at its turn, the second most targeted venue with 10 articles

Continents of the author(s). Figure 2.4 shows the distribution by continents of the authors of the primary studies. It is worth remarking here that some studies have authors from more than one continent. Specifically, 11 primary studies have authors from 2 continents, and 3 primary studies have authors from 3 continents. This is the reason why the total number of primary studies in the graph is greater than the number of primary studies of this SMAP. Authors are from countries all over the world, except from Oceania. There is a special focus on Europe (65 primary studies) and North America (25 primary studies).

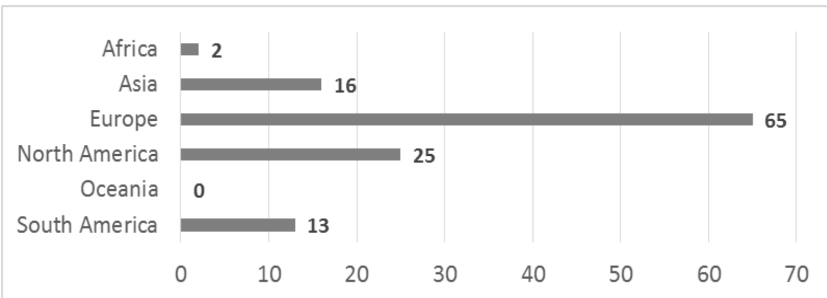


Figure 2.4 – Primary studies per continents of the authors

Table 2.6 – Countries with the biggest figure of primary studies

Rank	Country	Number of primary studies
1	USA	20
2	Spain	18
3	Germany	18
4	United Kingdom	10
5	Brazil	9

Looking at the distribution by countries, Table 2.6 shows the ones with the biggest figure of authors of the primary studies. It is worth remarking that when the authors of a primary study are from different countries, all the countries are counted as having a primary study. USA, Spain and Germany are the top 3 countries (with 20 primary studies the first one, and 18 the second and third ones); at its turn, Brazil and United Kingdom have 10 and 9 primary studies, respectively. The rest of countries from where authors of the primary studies are have less than five primary studies.

Overview of the Proposal

General classification of the proposals. The proposals were classified on a high level according to the type of requirements reuse strategy they propose, a mix of what is being reused and the structure used to reuse the knowledge. This classification, which was built while reading the primary studies, drove to six categories:

- Process Patterns proposals incorporate patterns to improve the RE processes in organizations, e.g. P25, which proposes RE Patterns of good RE practices.
- Writing Guidelines Patterns proposals deal with patterns that propose guidelines or rules to improve the writing of the requirements (either in natural or formal language) (e.g. P06, which proposes the use of Grammatical Knowledge Patterns, a king of restricted grammar, to write requirements.)
- Formalization/Specification Patterns proposals are related to patterns that, given a requirements specification (either in natural or formal language), convert this specification into a new one with a different format (e.g. P05, which presents a set of Source Patterns aimed to identify, from annotated business models, the functionalities and the relationships existing among them, and that serve as the basis to transform business models into use case diagrams).
- Domain/NF Patterns proposals present a set of requirements that is aimed to be reused completely for a specific domain or non-functional aspect (e.g. P08, which proposes to model provenance requirements with a modified version of the NFR Patterns proposed by Suppakkul [10]).
- Templates proposals offer a set of requirements (usually parameterized requirements) ready to be used (e.g. P12, which proposes the use of Requirements Boilerplates to aid requirements engineers in the formulation of security requirements).
- Patterns with Context Information proposals are a more advanced type of templates, where apart from the template per se, information about when to use the template and other data is provided (e.g. P77, which proposes Requirement Patterns that explain what requirements needs to convey, offer potential questions to ask, points out potential pitfalls, suggest extra requirements, and other advice).

Figure 2.5 shows the number of proposals per each one of the six identified categories. While reading the primary studies, some proposals were difficult to be placed in just one category, mostly because they are proposing different aspects or approaches to reuse inside the same proposal. That happens in the case of 9 proposals: P1, P2, P3, P11, P33, P40, P69, P74, and P76. This is the reason why the total number of proposals in the graph is greater than the number of proposals of this SMAP. There is a higher share of contributions that deal with patterns that have context information associated (33 proposals), followed by templates (21 proposals) and domain/NF patterns (16 proposals).

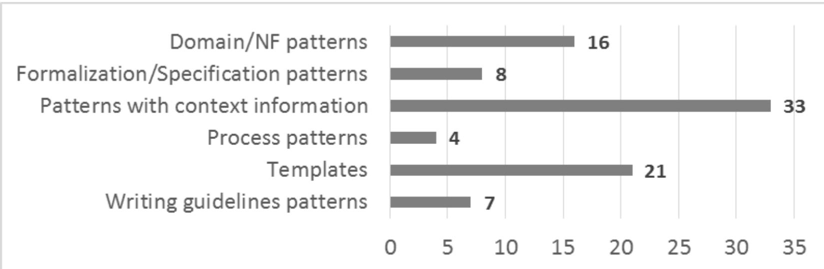


Figure 2.5 – Proposals per type of proposal

Scope of the proposals. Figure 2.6 shows the distribution of proposals according to the scope of the RE stages they are dealing with: the elicitation of requirements, the specification of requirements or both of them. As it might be observed, most of the proposals (59 out 80) deal with both the elicitation and specification of requirements, which makes sense when thinking that reusing knowledge could improve not only the gathering of the requirements but also their writing.

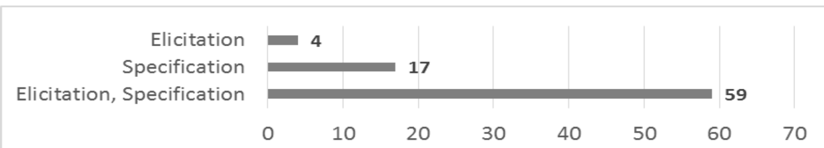


Figure 2.6 – Proposals per scope

The distribution of proposals according to the type of proposal and their scope is shown in Table 2.7. In the case of process patterns, they propose the improvement of practices in both the specification and elicitation of requirements. As it could be expected, writing guidelines patterns and formalization/specification patterns are never used alone for elicitation as their main goal is the specification of requirements, and they are used for both elicitation and specification on those proposals that incorporated various types of requirements reuse patterns

(P3, P40, and P76). Finally, domain/NF patterns, templates and patterns with context information are mostly, if not only, focused on both the elicitation and specification of requirements.

Table 2.7 – Scope by type of proposal

		Type of proposal					
		Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context information	Process patterns	Tem-plates	Writing guidelines patterns
Scope	Elicitation	3	0	1	0	0	0
	Specification	3	6	2	1	0	6
	Elicitation, Specification	10	2	30	3	21	1
TOTAL		16	8	33	4	21	7

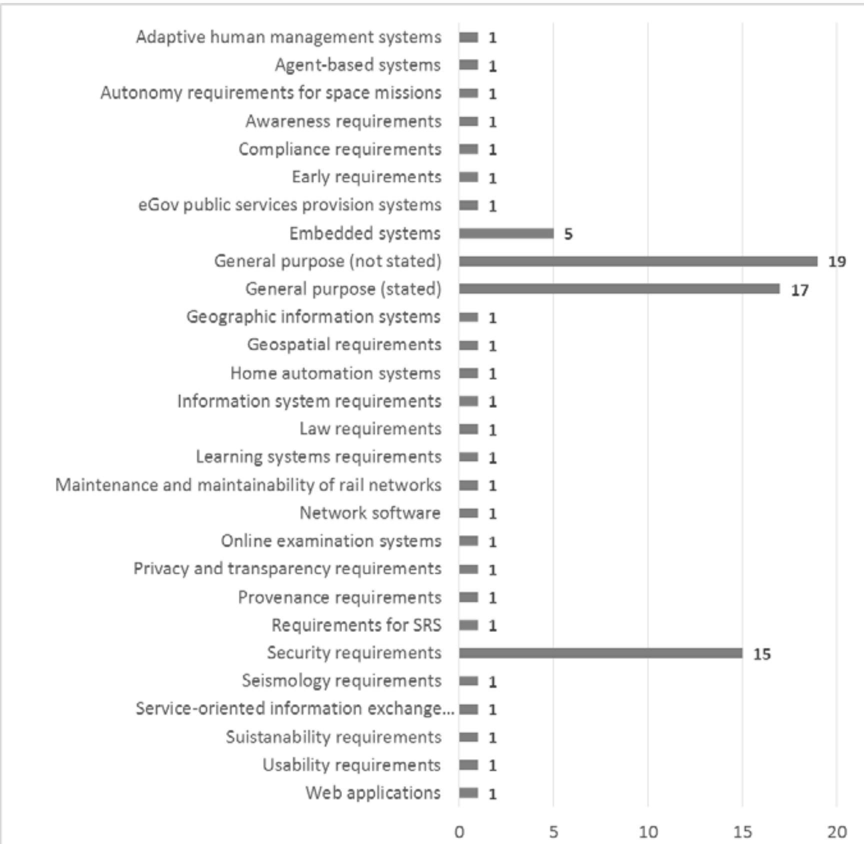


Figure 2.7 – Proposals per domain

Domain of the proposals. The proposal's domain corresponds to the field of information systems where the RE knowledge is being reused (Figure 2.7). On the one hand, 26 specific domains are targeted by the proposals, being security requirements the most recurrent one (15 proposals), which is not surprising since security requirements are considered very relevant in systems. On the other hand, there are also proposals that have a general purpose in respect to the domain. These general-purpose proposals were classified in two groups: there are proposals where the authors explicitly state the domain as general, but there are other proposals where the authors do not state a specific domain, and they were thought as generally applicable while reading the proposal. Finally, following this same line of general-purpose domain, some proposals, although stating a specific domain, could be easily applicable to any other domain; that is the case of proposals P39, P61, P66, and P71.

Table 2.8 shows the distribution of domains according to the type of proposals. In all the types of proposals, except for domain/NF patterns, general purpose is the most usual domain (5 out of 8 for formalization/specification patterns, 14 out of 33 for patterns with context information, 3 out of 4 for process patterns, 9 out of 21 for templates, and 5 out of 7 for writing guidelines patterns). Security requirements is also quite common in domain/NF patterns (4 out of 16), patterns with context information (6 out of 33), and templates (7 out of 21). It is also worth remarking that for domain/NF patterns and patterns with context information, there is a big ratio of proposals that are not of general purpose neither focused on security requirements (specifically, 9 out of 16 for domain/NF patterns and 13 for patterns with context information); these numbers correspond to the addition of the figures corresponding to domains that are not general neither security requirements.

Type of requirements reused by the proposals. Another aspect to take into account while reviewing the proposals was the type of requirements they pretend to reuse: FRs, NFRs and/or NTRs. The distribution of proposals according to the type of requirements they reuse is shown in Figure 2.8.

For some of the proposals, though, this aspect is not applicable; that is the case of the 4 proposals of process patterns, since this kind of proposals target to reuse best practices related to requirements instead of specific requirements. They are reflected in the graph as *Not Apply*.

Some proposals do not state the type of requirements they targeted to reuse, neither it was inferable while reading the papers. That occurs in the case of 8 proposals: P18, P20, P23, P26, P46, P52, P53, and P79. This fact is marked in the graph as *Not Stated*.

Table 2.8 – Domain by type of proposals *

		Type of proposal					
		Domain /NF patterns	Formali-zation/ Spec patterns	Patterns with context info	Process patterns	Tem-plates	Writing guide-lines patterns
Domain	Adaptive human management systems	1	0	0	0	0	0
	Agent-based systems	1	0	0	0	0	0
	Autonomy requirements for space missions	1	0	0	0	1	0
	Awareness requirements	1	0	0	0	0	0
	Compliance requirements	0	1	0	0	0	0
	Early requirements	1	0	0	0	0	0
	eGov public services provision syst	1	0	0	0	1	0
	Embedded systems	0	1	3	0	1	1
	General purpose (not stated)	0	**3**	**8**	**1**	**6**	**1**
	General purpose (stated)	3	**2**	**6**	**2**	**3**	**4**
	Geographic information systems	0	0	1	0	0	0
	Geospatial requirements	0	0	1	0	0	0
	Home automation systems	0	0	0	0	1	0
	Information system requirements	0	0	1	0	0	0
	Law requirements	0	0	1	0	0	0
	Learning systems requirements	0	0	1	0	0	0
	Maintenance and maintainability of rail networks	0	0	0	0	1	0
	Network software	0	0	0	0	0	1
	Online examination systems	0	0	1	0	0	0
	Privacy and transparency reqs.	1	0	0	0	0	0
	Provenance requirements	1	0	0	0	0	0
	Requirements for SRSs	0	0	0	1	0	0
	Security requirements	**4**	1	**6**	0	**7**	0
	Seismology requirements	0	0	1	0	0	0
	Service-oriented information exchange requirements	1	0	0	0	0	0
	Sustainability requirements	0	0	1	0	0	0
	Usability requirements	0	0	1	0	0	0
	Web applications	0	0	1	0	0	0
	TOTAL	16	8	33	4	21	7

* The biggest figures per each type of proposal are highlighted with a different background colour

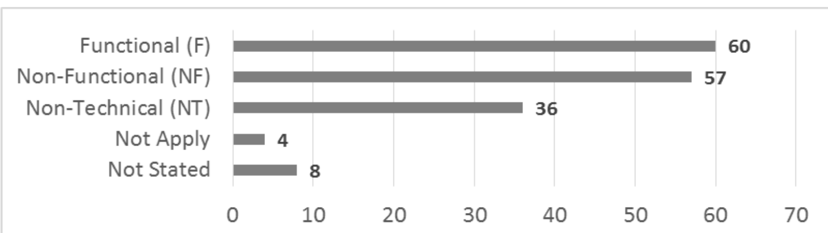


Figure 2.8 – Proposals per type of requirements being reused

Regarding the proposals that specify the type of requirement being reused, some of them are dealing with more than one type at the same time. Specifically, 15 proposals are dealing with 2 types, while 35 proposals are dealing with 3 types (i.e. all types of requirements); that is the reason why the sum of all the figures on the graph is greater than the number of proposals of this SMAP. The figures reflect that most of the proposals are dealing with FRs (60 out of 80 proposals) and NFRs (57 out of 80 proposals). On the contrary, only almost a half of the proposals are dealing with NTRs (36 out of 80 proposals). The reason behind that could be that NTRs are not always included in software projects, or when they are included, they do so as part of NFRs.

The distribution of type of requirements by type of proposals is presented in Table 2.9. For patterns with context information, templates and writing guidelines patterns, FRs and NFRs are dealt with by the majority of proposals (32 for FRs and 28 for NFRs out of 33 for patterns with context information, 14 for FRs and 15 for NFRs out of 21 for templates, and 6 for FRs and 5 for NFRs out of 7 for writing guidelines patterns). In the case of domain/NF patterns, the proposals are dealing mostly with NFRs (14 out of 16), while in the case of formalization/specification patterns they are dealing mostly with FRs (5 out of 8). As stated before, for process patterns the type of requirement is not applicable.

Table 2.9 – Requirement types by type of proposals *

		Type of proposal					
		Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context info	Process patterns	Tem-plates	Writing guidelines patterns
Requirement types	FRs	10	**5**	**32**	0	**14**	**6**
	NFRs	**14**	3	**28**	0	**15**	**5**
	NTRs	7	3	18	0	10	4
	Not Apply	0	0	0	**4**	1	0
	Not Stated	1	3	0	0	3	1

* The biggest figures per each type of proposal are highlighted with a different background colour

Artefact to reuse by the proposals. The artefacts that are reused by the proposals are shown in Figure 2.9. Again, as an only proposal sometimes reuse more than one artefact, the sum of the figures in the graph is bigger than the total of proposals of this SMAP. Specifically, 17 proposals are reusing 2 types of artefacts, 2 proposals 3 types of artefacts, 3 proposals 4 types of artefacts, and 1 proposal 6 types of artefacts. The most reused asset is Sentences in Natural Language (SNL) that correspond to explicit requirements (50 out of 80 proposals). This fact is not surprising taking into account that natural language is still considered the most prominent form to specify requirements, as corroborated by Weston et al. [107]. The second position is taken by use cases (13 proposals), followed by conceptual models (6 proposals), refinement models (5 proposals) and formalization of requirements (4 proposals). Other well-known assets like activity diagrams, sequence diagrams and misuse cases are not appearing in more than 3 proposals. One of the proposals (P1) states that the artefact to reuse vary from template to template.

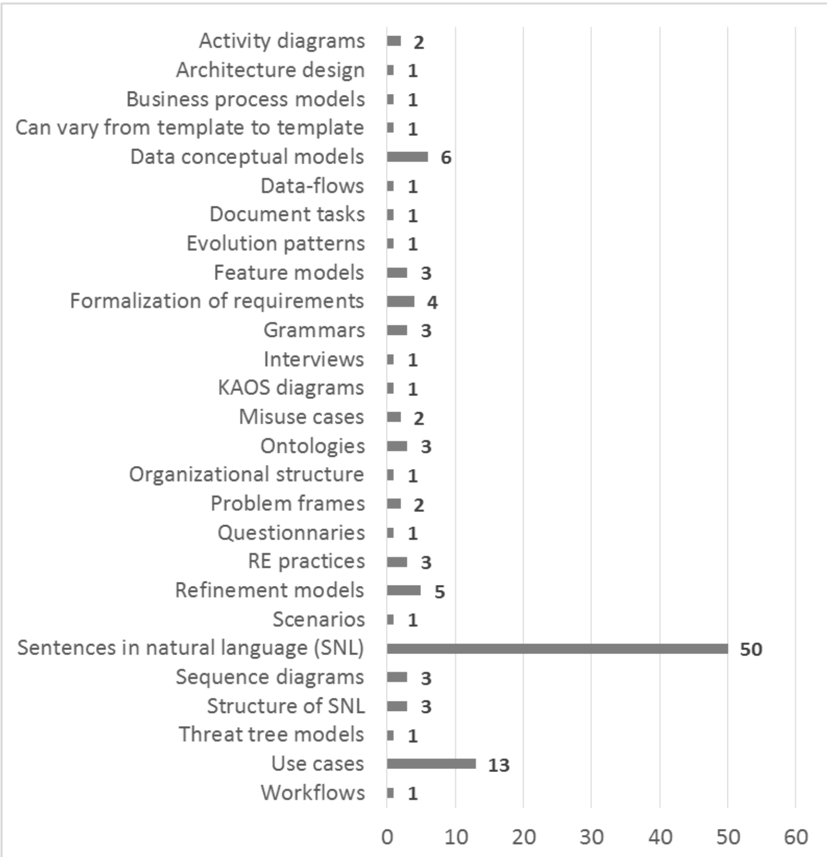


Figure 2.9 – Proposals per artefact to reuse

In that sense, it is interesting to study how the artefacts being reused are spread among the specific types of proposals (Table 2.10). The biggest share of proposals corresponding to patterns with context information and templates are mostly reusing SNL (29 out of 31 and 18 out of 23 proposals, respectively). These figures also reflect the fact that SNL are mostly being reused by these two types of proposals. In the case of domain/NF patterns, SNL also hold a prominent position, although in this case with a smaller share of proposals, only 6 out of the

16; in this same type of proposal, SNL is closely followed by *i** refinement models and use cases (4 out of 16 proposals in each case). In the case of formalization/specification patterns, the formalization of requirements gets the biggest figure (3 out of 8 proposals), being the other reused artefacts SNL, activity diagrams, evolution patterns, and grammars. Finally, most of the process patterns proposals are reusing RE practices (3 out of 4 proposals), and most of the writing guidelines patterns are reusing either grammars or the structure of SNL (3 out of 7 proposals for each artefact).

Table 2.10 – Artefact to reuse by type of proposals *

Artefact to reuse	Type of proposal: Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context info	Process patterns	Tem-plates	Writing guidelines patterns
Activity diagrams	0	1	0	0	1	0
Architecture design	1	0	0	0	0	0
Business process models	0	0	1	0	0	0
Can vary from template to template	0	0	0	1	1	0
Data conceptual models	1	0	5	0	0	0
Data-flows	0	0	0	0	1	0
Document tasks	0	0	0	0	1	0
Evolution patterns	0	1	0	0	0	0
Feature models	0	0	3	0	0	0
Formalization of reqs.	1	**3**	0	0	1	0
Grammars	0	1	0	0	0	**3**
*i** refinement models	**4**	0	0	1	0	0
Interviews	0	0	1	0	0	0
KAOS diagrams	1	0	0	0	0	0
Misuse cases	0	0	1	0	1	0
Ontologies	1	0	1	0	2	0
Organizational structure	1	0	0	0	0	0
Problem frames	2	0	0	0	0	0
Questionnaires	0	0	1	0	0	0
RE practices	0	0	0	**3**	1	0
Scenarios	0	0	0	0	1	0
Sequence diagrams	0	0	3	0	0	0
Sentences in natural language (SNL)	**6**	2	**29**	0	**18**	1
Structure of SNL	0	0	0	0	1	3
Threat tree models	1	0	0	0	0	0
Use cases	**4**	1	6	0	3	0
Workflows	0	0	0	0	1	0

* The biggest figures per each type of proposal are highlighted with a different background colour

Notation used to define the artefact to reuse by the proposals. Apart from the artefact being reused, the notation used to specify this object was also analysed. Figure 2.10 shows that different requirements notations are tackled, which represents the inherent heterogeneity of requirement formats. As some proposals are dealing with more than one artefact to reuse RE knowledge, it is also the case that some proposals are dealing with more than one notation. This is the reason why the number of proposals in the graph is greater than the number of proposals of this SMAP. Concretely, 25 proposals use 2 types of notations, and only 1 proposal 3 types of notations. It is not surprising that, since most of the proposals are reusing SNL, the notation natural language gets the highest figure (63 out of 80 proposals). This notation is followed in the rank by UML (11 proposals), *i** (7 proposals), and temporal formal logic (5 proposals). The other notations are not used by more than 3 of the proposals. It is worth remarking that some of the proposals specify that they are using, for instance, some kind of formal language, but they do not specifically state what formal language is; this is the case of 1 proposal for formal languages, 1 proposal for variability models languages, 2 proposals for modelling languages, 2 proposals for ontology languages, and 3 proposals for grammar languages. In addition, 2 proposals do not specify any type of notation used to specify the artefact to reuse (P31, P36), and one of them states that it could vary from artefact to artefact (P1).

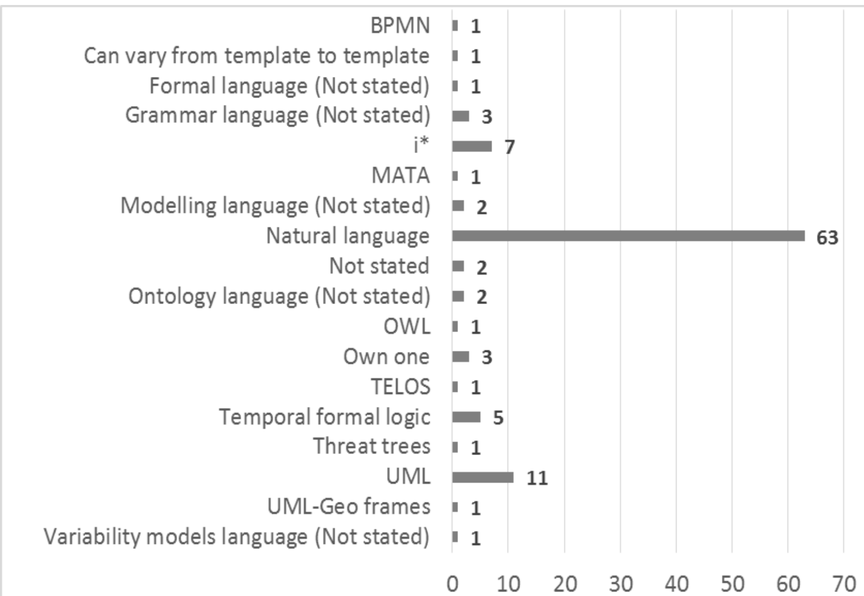


Figure 2.10 – Proposals per notation used to define the artefact to reuse

Table 2.11 contains the distribution of the notation languages used in the artefacts to reuse according to the type of proposals. As can be seen, natural language gets the biggest share of proposals in all the types (8 out of 16 for domain/NF patterns, 4 out of 8 for formalization/specification patterns, 30 out of 33 for patterns with context information, 3 out of 4 for process patterns, 20 out of 21 for templates, and 7 out of 7 for writing guidelines patterns). In the case of domain/NF patterns, *i** and UML are quite common too (6 and 4 out of 16, respectively). As for formalization/specification patterns, temporal formal logic is also fairly used (3 out 8). The same situation appears in writing guidelines patterns proposals using grammar languages (3 out of 7). Finally, although it does not represent a big percentage, UML takes the second position when looking at patterns with context information proposals (5 out of 33).

Table 2.11 – Notation language by type of proposals *

		Type of proposal					
		Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context info	Process patterns	Tem-plates	Writing guidelines patterns
Notation language	BPMN	0	1	0	0	0	0
	Can vary from template to template	0	0	0	1	1	0
	Formal language (Not stated)	0	1	0	0	1	0
	Grammar language (Not stated)	0	1	0	0	0	**3**
	*i**	**6**	0	0	1	0	0
	MATA	0	0	1	0	0	0
	Modelling language (Not stated)	0	0	2	0	0	0
	Natural language	**8**	**4**	**30**	**3**	**20**	**7**
	Not stated	0	0	1	0	1	0
	Ontology language (Not stated)	1	0	1	0	1	0
	OWL	0	0	0	0	1	0
	Own one	0	2	1	0	0	0
	TELOS	1	0	0	0	0	0
	Temporal formal logic	2	**3**	1	0	0	0
	Threat trees	1	0	0	0	0	0
	UML	**4**	2	**5**	0	1	0
	UML-Geo frames	0	0	1	0	0	0
	Variability models language (Not stated)	0	0	1	0	0	0

* The biggest figures per each type of proposal are highlighted with a different background colour

Structure

Structure in which the artefact to reuse is contained. Looking at the primary studies, it was easy to realize that the artefacts being reused are not isolated, but they are accompanied by other information that could help the reuse process, such as the artefacts goal, the artefacts description or even information useful for other stages that applies when reusing the artefact. When that happens, it is considered that the artefact to be reused is contained in a bigger structure. In the case that there is not any extra information coming with the artefact, not even metadata, it is considered to be isolated (i.e. not contained in a bigger structure). The distribution of the proposals according to this aspect is shown in Figure 2.11. There are 56 out of the 80 proposals where the artefact to be reused is classified as being part of a bigger structure, in contrast to 24 proposals where the artefact to be reused is considered an isolated unit of knowledge.

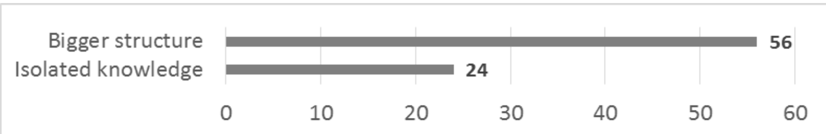


Figure 2.11 – Proposals per structure in which the artefact to reuse is contained

Table 2.12 – Structure in which the artefact to reuse is contained by type of proposal

		Type of proposal					
		Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context information	Process patterns	Templates	Writing guidelines patterns
Bigger structure	No	3	2	0	0	19	5
	Yes	13	6	33	4	2	2
TOTAL		16	8	33	4	21	7

Looking at the distribution of the types of proposals according to the fact if the artefact to be reused is contained in a bigger structure or not (Table 2.12), in most of the proposals the artefact to be reused is part of a bigger structure when the proposal is classified as domain/NF patterns, formalization/specification patterns, patterns with context information, and process patterns. This result could be justified by the fact that having extra information ease the reuse of RE knowledge, like for instance a name that reflects what is being reused, a short description, or even the reasoning for having arrived to the knowledge being reused.

In the case of proposals corresponding to the templates and writing guidelines patterns categories, the contrary is happening: the artefacts to be reused in most of the proposals in these categories is not surrounded by any extra information. In the case of templates, this fact is not surprising since, by definition, this SMAP has classified as templates the proposals that offer ready-to-use requirements, but that do not offer any extra information. In the case of writing guidelines patterns, the justification is related to fact that these proposals usually just offer a set of guidelines or structure for the requirements sentences or RE knowledge, and in that context there is no too much extra information that could be offered to the end-user to ease the reuse process.

Structure information. At this point, the most occurring attributes that accompany the artefact to be reused were analysed (Figure 2.12). These attributes are: metadata (like name, description, problem solved, etc.); applicability conditions (i.e. if there is an explanation about in what conditions it is appropriate to reuse the artefact); parameters (if the artefact to be reused contains some parameters that need to be filled when reusing it); and finally, other stages information (referring to useful information for further development stages associated to the artefact being reused, like code considerations, tests, etc.).

Metadata, applicability conditions and other stages information are possible to appear only in those proposals where the artefact to be reused is contained on a bigger structure (56 proposals). Taking that into account, most of the proposals that consider a bigger structure (56 proposals, as explained in the previous aspect) commonly include also metadata (45 proposals) and applicability conditions (39 proposals). The other stages information is a less common attribute (only 14 proposals contain it). In the case of parameters, they are applicable to all the proposals (not taking into account if the artefact to be reused is included in a bigger structure or not), since it is an attribute included in the artefact to be reused. Only a quarter of the proposals (22 out of 80) include parameters, which is quite surprising when thinking that parameters is quite an easy way to abstract the knowledge being reused.

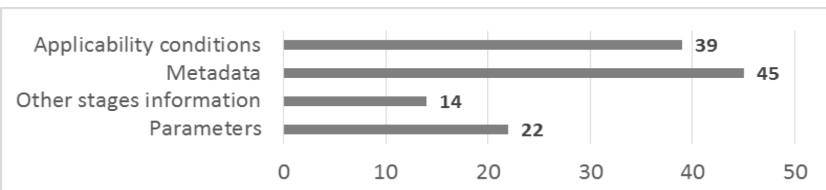


Figure 2.12 – Proposals per information accompanying the artefact to reuse

Organization

Relationships. Some of the artefacts to reuse present several types of relationships among them (Figure 2.13). Specifically, there are 38 proposals that incorporate relationships.

These relationships help when reusing knowledge, since they state what other artefacts may, for instance, help or contradict the already reused artefact. Analysing the proposals, most of them present different types of relationships. Just to present some of them, P9 incorporates the *generalization* and *specialization* relationships; P59, P68 and P80 go a step further and apart from the *specialization-of*, also use the *part-of* and *occurrence-of* relationships. P13 deals with *weakening* and *strengthening* relationships, and P21 incorporates these two types too (but in their case naming them *damages* and *contributes*), but it has also *implies* and *excludes* relationships. P77 presents more detailed types of relationships, that may be purely structural like *has*, *uses* and *is-a*, or with a semantic meaning like *displays* and *is across*.

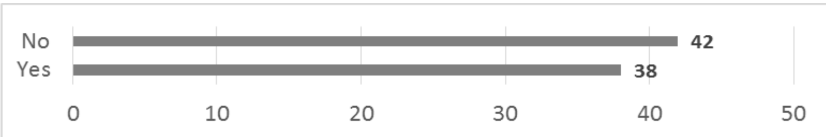


Figure 2.13 – Proposals per relationships among artefacts to reuse

Arrangement. From a reuse point of view, it was also interesting to investigate how the artefacts to reuse were organized, if that was the case, in a higher level (Figure 2.14). The fact of having the artefacts to reuse organized or indexed somehow facilitates their access, and therefore their reuse. In that sense, 12 proposals presented a taxonomy for organizing the artefacts (i.e. a set of words that have been organized to control the use of terms to facilitate the storing and retrieving of items), and 29 proposals a repository with some classification. However, 39 proposals did not state any kind of arrangement.

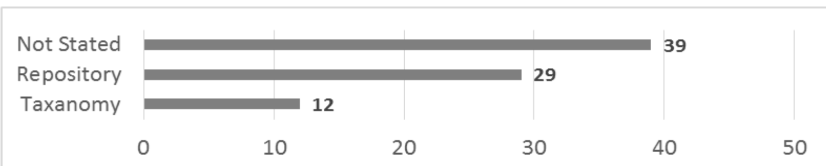


Figure 2.14 – Proposals per arrangement of the artefacts to reuse

Classification. From the proposals that present a specific type of arrangement of their artefacts to reuse, some of them go a step further and also propose a specific classification for either their repository or taxonomy. The distribution of the proposals according to the fact whether they presented a specific classification or not is shown in Figure 2.15. From the 41 proposals presenting a taxonomy or repository, 19 of them include a specific classification to organize the artefacts to reuse, and 22 of them do not do so. The 39 proposals not stating a classification correspond to the 39 proposals not presenting any kind of arrangement of their artefacts to reuse.

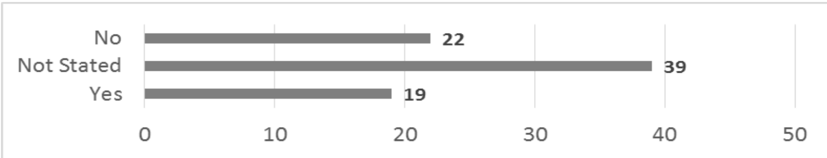


Figure 2.15 – Proposals per specific classification proposed for the artefacts to reuse

Metamodels

Metamodel for the artefact to reuse. Some additional assets apart from the core of the proposal, which is the artefact to reuse, were also analysed. One of these assets is the metamodel that specifies the artefact to reuse. Having such kind of metamodel implies that the artefact to reuse probably underwent through a deeper analysis, construction and development; that it incorporates rules, constraints, and theories applicable and useful. Taking that into account, what is surprising is that only 17 proposals state to have a metamodel for the artefact to reuse (Figure 2.16).

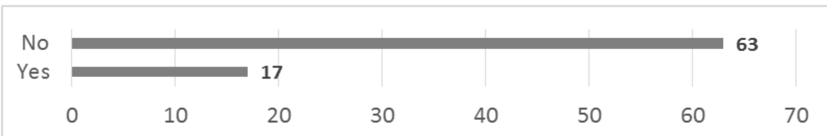


Figure 2.16 – Proposals per metamodel for the artefact to reuse

Metamodel for the arrangement. Another asset for which it could be interesting to have a metamodel is the arrangement of the artefacts to reuse. The same reasoning as for having a metamodel for the artefact to reuse is applicable here: a metamodel for the arrangement probably implies a better thinking of how the artefacts to reuse are organized. Figure 2.17 shows the distribution of proposals according if they incorporate such a kind of metamodel. Surprisingly, only 3 proposals incorporate it, those being P19, P21 and P61.

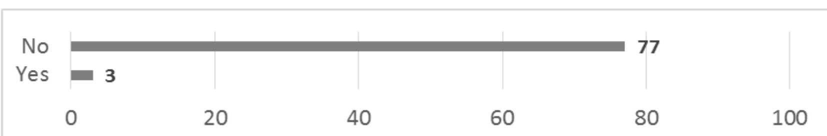


Figure 2.17 – Proposals per metamodel for the arrangement of the artefacts to reuse

Methods

Method to construct the artefacts to reuse. This point makes reference to the existence of a specific method presented in the proposals in order to construct the artefacts to reuse. As shown in Figure 2.18, most of the proposals do not consider such a method (47 out of 80 proposals). Contrarily, 21 proposals present a well-structured method to construct their artefacts to reuse, while 12 proposals incorporate only some guidelines for the construction (shown as *Partial* in the graph).

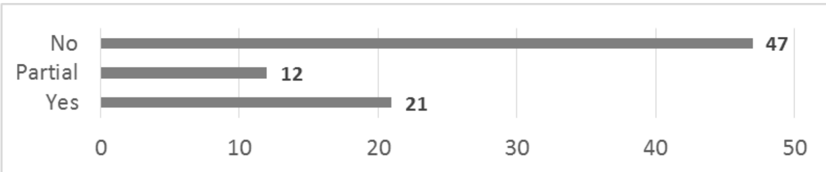


Figure 2.18 – Proposals per method to construct the artefact to reuse

Method to reuse the artefacts. Having a method to reuse the artefacts is probably as important as having an artefact to reuse. Without a method, it could be difficult to know how to properly reuse the artefacts, as well as how to incorporate the approach in a real environment. That is corroborated by the fact that this type of method is the most common in the proposals: 47 out of 80 proposals present some kind of method to reuse their artefacts (see Figure 2.19); specifically, 30 of them propose a well-structured method to do the reuse process, and 17 of them propose at least some guidelines to do it.

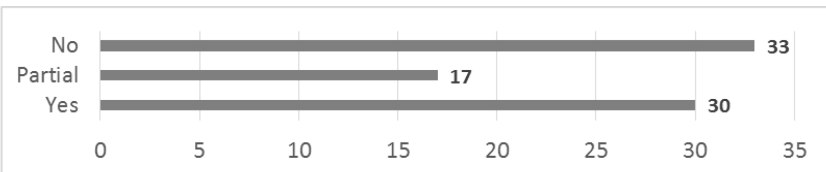


Figure 2.19 – Proposals per method to reuse the artefacts

Method to update the catalogue of artefacts to reuse. The last method analysed in the proposals is the one related to the evolution of the catalogue of artefacts to reuse, i.e. a method to maintain the catalogue up-to-date, usually with information about the actual use of the artefacts in a real environment. Figure 2.20 shows the distribution of the proposals according of the existence of such a method. While the majority of proposals do not include an update method (65 proposals out of 80), there are 11 proposals that present some guidelines for the update (labelled as *Partial* in the graph) and 4 proposals that present a complete method. These proposals are P14, P21, P30 and P49. Finally, it is worth remarking that one proposal do not include neither a method nor guidelines for update, but they mention the proposal manages the evolution of the artefacts to reuse. (P22).

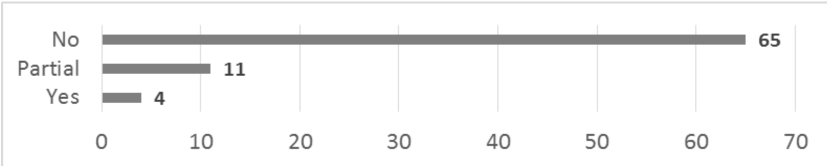


Figure 2.20 – Proposals per method to update the catalogue of artefacts to reuse

Catalogue

Existence of a catalogue of artefacts to reuse. A catalogue of artefacts to reuse consists on a set of ready-to-use reusable artefacts. The distribution of proposals according to the existence of a catalogue is presented in Figure 2.21. Most of the proposals (63 out of 80) actually include some kind of catalogue.

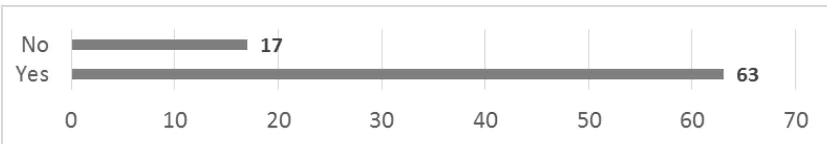


Figure 2.21 – Proposals per existence of a catalogue of artefacts to reuse

Table 2.13 – Existence of a catalogue of artefacts to reuse by type of proposal

		Type of proposal					
		Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context information	Process patterns	Templates	Writing guidelines patterns
Catalogue existence	No	2	1	8	0	4	2
	Yes	14	7	25	4	17	5
TOTAL		16	8	33	4	21	7

Table 2.13 contains how the proposals are spread according to the type of proposal and the existence of a catalogue of artefacts to reuse. As seen there, there are catalogues of ready-to-use artefacts in all types of proposals. Looking at the distribution of each type of proposal, domain/NF patterns, formalization/specification patterns and process patterns are the ones with a higher rates of catalogues (14 out of 16, 7 out 8, and 4 out of 4, respectively), all of them with a partial percentage higher than 87'5%. For patterns with context information, templates

and writing guidelines patterns, this percentage is between 70% and 80% (25 out of 33, 17 out 21, and 5 out of 7, respectively).

Domain of the catalogue of artefacts to reuse. Figure 2.22 contains the distribution of the proposals according to the domain of the catalogue of artefacts to reuse. By domain it is meant if the catalogue is specific for a project (like a test, a case study or so), or either it is general and could be used in any other environment. For 17 proposals this aspect does not apply since they do not incorporate any catalogue. While 25 proposals have a catalogue for a specific project, for the majority of the proposals the catalogue is general (37 proposals). For one proposal that include a catalogue, it was not possible to determine if it is specific or general (P73), which is labelled as *Not Stated* in the figure.

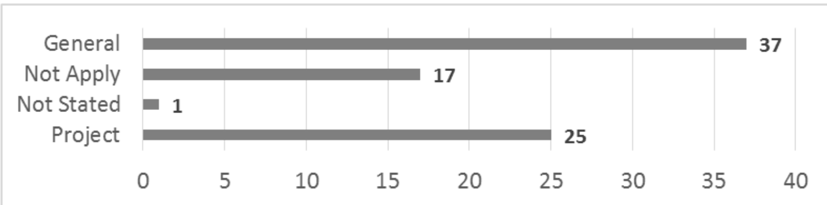


Figure 2.22 – Proposals per domain of the catalogue of artefacts to reuse

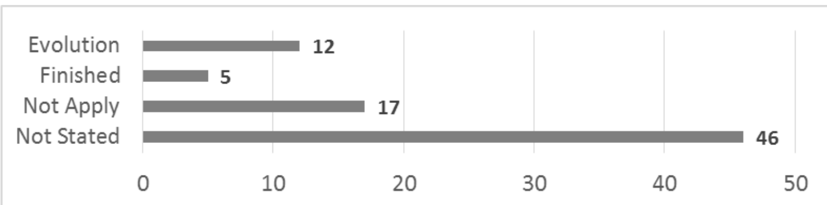


Figure 2.23 – Proposals per state of the catalogue of artefacts to reuse

Table 2.14 – State of the catalogue of artefacts to reuse by its domain

		State				TOTAL
		Evolution	Finished	Not Apply	Not Stated	
Domain	General	9	4	0	24	37
	Not Apply	0	0	17	0	17
	Not Stated	1	0	0	0	1
	Project	2	1	0	22	25
TOTAL		12	5	17	46	80

State of the catalogue of artefacts to reuse. The state of the catalogue makes references to whether it is considered to be either finished or still under evolution (i.e. in construction). The distribution of the proposals by the state of the catalogue is presented in Figure 2.23. As happens with the catalogue domain, the state of the catalogue is not applicable for those proposals that do not incorporate any catalogue (17 proposals). Only 5 proposals state to have a finished catalogue, in contrast to the 12 proposals having a catalogue that is still in evolution. Unfortunately, for most of the proposals (46 proposals) it was not possible to determine what the state of the catalogue is.

In Table 2.14 it is possible to see the cross-tabulation of the proposals according to the state of the catalogue and its domain. The remarkable fact here is that only 4 proposals are including a catalogue that is general and finished, so out of 80, only 4 of them have a catalogue that is complete and ready-to-use. These proposals are P21, P32, P53, and P70.

Tools and Validation

Tools. Figure 2.24 contains the number of proposals that either have tools to facilitate the reuse process or not. Only 28 proposals out of 80 incorporate such kind of tools. This indicates that most proposals are, unfortunately, not tool supported. A possible explanation could be that in the academic environment where these proposals were created, tool implementation is not the main focus. In order to increase the chances of a proposal being adopted in practice, suitable tools must be available.

Overall, these tools are used in the proposals to model and store the specific reusable knowledge of each proposal (i.e. the artefacts to reuse), and also to visualize such knowledge so it can be converted with these tools in reused knowledge (i.e. the tools help to facilitate the reuse process of the artefacts to reuse). Although the general trend is proposing new tools built from scratch, 3 proposals implement these functionalities as plugins of commercial tools: in P16 the commercial tools are IBM Rational Doors and Pure Variants, in P47 they are IBM Rational Doors and Excel, and in P49 it is Requisite Pro.

When looking at the proposals including tools by their type of proposal (Table 2.15), they are evenly distributed between the types. For all the types except domain/NF patterns and writing guidelines patterns, the specific percentage of proposals including tools for each type is in the rank 40%-50%. For domain/NF patterns and writing guidelines patterns this percentage is 19% and 29%, respectively. This could be caused by the fact that these types of proposals usually include either lists of artefacts to be reused as they are, in the case of domain/NF patterns, or general writing guidelines, in the case of writing guidelines patterns, so the authors do not consider it is necessary to develop a specific tool to be able to reuse these assets.

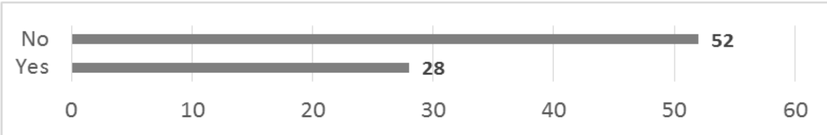


Figure 2.24 – Proposals per existence of tools

Table 2.15 – Existence of tools by type of proposal

		Type of proposal					
		Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context information	Process patterns	Templates	Writing guidelines patterns
Tools	No	13	5	21	2	11	5
	Yes	3	3	12	2	10	2
TOTAL		16	8	33	4	21	7

Tests. Another point worth looking at in the proposals was if they are tested in real environments (Figure 2.25). Almost half of the proposals, 31 out of 80, are tested. However, the data reported in the proposals reveals that the majority of proposals rely on toy examples or superficial case studies. Therefore, claims regarding the suitability of the proposals are supported by weak evidence, based mainly on the authors' own experience using the methods by means of informal case studies.

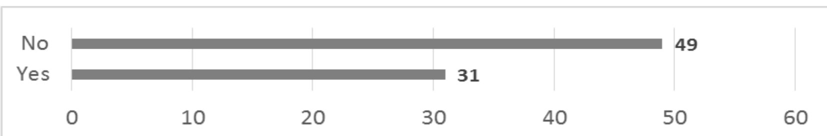


Figure 2.25 – Proposals per tests done

Table 2.16 shows the distribution of the types of proposal according to whether they are tested or not. Except in the case of domain/NF patterns and formalization/ specification patterns, the rate of tested proposals is around 40%-50% (being the minimum 40% and the maximum 57%). In the case of domain/NF patterns and formalization/specification patterns, however, this percentage was around 25%.

The relationship between proposals having tools and proposals being tested was also analysed (Table 2.17). The reasoning behind the analysis is that probably those proposals having tools would be the ones being tested (as having tools facilitates the test process).

Although, that is not the case. As the table shows, almost half of the proposals having tools are tested, while the other half are not. For proposals not having tools, almost the same distribution appears: 35% of the proposals with no tools re tested, and the other 65% are not.

Table 2.16 – Tests done by type of proposal

		Type of proposal					
		Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context information	Process patterns	Templates	Writing guidelines patterns
Proposal tested	No	12	6	20	2	11	3
	Yes	4	2	13	2	10	4
TOTAL		16	8	33	4	21	7

Table 2.17 – Tests done by existence of tools

		Tools		TOTAL
		No	Yes	
Proposal tested	No	24	15	49
	Yes	18	13	31
TOTAL		52	28	89

Type of research. The last aspect analysed in the proposals was the type of study they represent, dividing the proposals in two categories: empirical (proposals gaining knowledge by means of direct and indirect observation or experience), and non-empirical (not based on evidence from the real world). From empirical studies, it was found that experience reports (empirical studies where the authors have direct personal participation or observation) were an important share of them. This is the reason why it was decided to separate experience reports from the rest of empirical studies.

Figure 2.26 shows the number of proposals for each one of the previous categories. While most of the shares of the proposals are non-empirical (46 out of 80), there are 4 experience reports and 30 empirical studies different form experience reports. The data shows that extensive experimentation is needed to prove the validity of the proposals, to increase the adoption of proposals in industrial settings. This is why the rigor of empirical validation should be improved, presenting reliable arguments to demonstrate the methods adequacy to a particular situation. Specifically, there are not enough experience reports showing that requirement patterns can help everybody. Two major reasons for this could be the lack of access to requirement pattern catalogues and the late growth trend of requirement patterns.

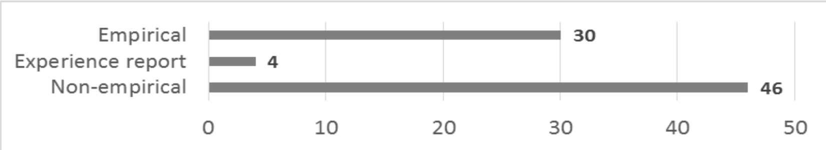


Figure 2.26 – Proposals per type of research

Table 2.18 – Type of research by type of proposal

Type of research		Type of proposal: Domain/ NF patterns	Formalization/ Specification patterns	Patterns with context information	Process patterns	Tem-plates	Writing guidelines patterns
Type of research	Empirical	4	3	12	1	8	5
	Exp. report	0	0	2	0	2	0
	Non-empirical	12	5	19	3	11	2
TOTAL		16	8	33	4	21	7

Table 2.19 – Type of research by tests done

		Proposal tested: No	Yes	TOTAL
Type of research	Empirical	8	22	30
	Experience report	0	4	4
	Non-empirical	41	5	46
TOTAL		49	31	80

Table 2.18 contains the distribution of proposals according to their type of proposal and their type of research. In the case of experience reports, their type of proposal is either patterns with context information or templates. The reasoning behind this is that, as these two types are the ones that are being most researched (both have the largest number of shares), they are also the ones that are actually being used in real organizations and real environments, thus producing the experience report research. Regarding the distribution of empirical studies, it is worth remarking that for writing guidelines patterns, 5 out of 7 proposals (representing 71%) are empirical studies. The percentage is much bigger when compared with the rest of types of proposals: 37% for formalization/specification patterns, patterns with context information and templates, and 25% for domain/NF patterns and process patterns.

Finally, the numbers of proposals being tested according to the type of research is shown in Table 2.19. As it could be expected, all experience reports are tested. While the majority of empirical studies are tested, 8 out of 30 are not. These eight proposals correspond to P05, P07, P09, P24, P34, P35, P43, and P65. The reason behind that is that the empirical studies presented in the proposals do not correspond to tests of the proposals. On the other hand, there are 5 proposals that correspond to non-empirical research but the authors claim the proposal is tested, although no data about the tests is provided (P25, P27, P41, P46, and P71).

2.2.4.2 Threats to Validity

In this part, all the aspects during the research process that might lead to a threat to validity, as well as the actions performed in order to mitigate them, are discussed. In this regard, the threats were identified and evaluated following Wohlin's et al. common classification of construct validity, internal validity, external validity and conclusion validity [108].

Construct Validity

Construct validity refers to the validity threats with respect to the observations performed in the study and if they really represent what is being investigated.

At this respect, one of the inherent threats to any systematic mapping is that it does not guarantee the inclusion of all the relevant works in the field. This might be caused by several reasons, for instance, a relevant work may not be indexed on the selected database, the keywords, title or abstract of a relevant work do not match with the keywords of the search, etc. This threat was mitigated by combining several databases (ACM Digital Library, IEEE Xplore, Science Direct and Springer Link) that include peer-reviewed journal and conference papers on RE, and complementing it with backward snowballing of all the primary studies after full-text reading. What is more, after having conducted the search phase, the researchers conducting the study (the applicant and one of her advisors) became aware of the papers of Braga et al. [98] and Mahendra and Ghazaryan [109], which present a systematic review on requirements reuse and a survey of requirement patterns in RE, respectively. These works review altogether 72 papers, where six additional papers were also relevant to this SMAP. These papers were added during the step of the protocol concerning snowballing.

Another potential risk that may lead to relevant papers being missed is the lack of agreed terminology in patterns for RE knowledge. This means that the choice of keywords may not have encompassed the complete set of papers published in the field of interest (i.e. patterns for RE knowledge used during the elicitation and specification stages). Indeed, as mentioned by Dybå and Dingsøyr [110], "it is important to recognize that software engineering keywords are not standardized and that they can be both discipline –

and language – specific". Hence, even carefully defining consistent keywords and related terms, there is a risk that pertinent studies are omitted. To minimize this risk, the terms used in the search string were broader than the actual scope of the SMAP (e.g. using "reuse" instead of "pattern" and all the possible synonyms of pattern).

The study was conducted at the beginning of 2015, and the report was wrote at the end of the same year. Therefore, studies from 2014 and earlier were included only in the analysis at first. It was contemplated whether also to include 2015 in the analysis and conduct the process for these studies as well. From a sampling perspective, it was important to have a good representation of studies. A total of 98 primary studies had been already identified, which were covering the different areas were requirement patterns had been used. Furthermore, different types of publication venues were already well represented. At the end, it was decided to explore manually the most important RE journals and conferences for 2015 (Requirements Engineering Journal, the IEEE Requirements Engineering Conference, and the International Working Conference on Requirements: Foundation for Software Quality), thus assuring a good coverage for 2015 too.

Researcher bias may appear during the selection of studies. To reduce this threat and to gain confidence in the results, the selection was conducted by two researchers.

However, the construct validity may not be completely solved since the problem goes beyond an accurate protocol. On the one hand, it also concerns issues related to the targeted papers (e.g. not mentioning relevant keywords explicitly or having inaccurate abstracts). On the other hand, the automatically retrieved results rely on the functionality and precision of the search engines of the used digital libraries, but unfortunately, many search engines of computer science digital libraries turned out to be unreliable.

Internal Validity

Internal validity is related to the validity of the analysis performed. Concerning this aspect, three major threats were identified.

To reduce the threat of describing accurately the observations extracted from the primary studies, a data collection form (composed of different aspects) was designed to support the recording of data. The form objectified the data extraction process and could always be revisited.

With respect to bias in the data extraction, some difficulties were found to extract relevant information from the papers. Some papers do not present objective details regarding some of the aspects addressed in the data extraction form. For instance, several papers do not explicitly mention what requirements phase the proposal is covering. This situation poses a challenge when analysing the different aspects, and a subsequent threat to validity. For those aspects that were not clearly stated in the proposals, if it was possible for the authors to interpret the subjective information provided by the papers and to abstract the value of the aspect from such information, a specific value to the aspect was given. To minimize interpretation bias on these aspects, discussion meetings were held among the researchers conducting the study during the data extraction phase. For those aspects whose values were not clearly stated or impossible to abstract from the information presented in the papers, they were given the value of *not sated* if the aspect corresponded to a characteristic of the proposals or *not included* (i.e. value *no*) when the aspects corresponded to an asset of the proposal.

During the data extraction phase, researcher bias is also a threat. This threat may arise while doing the full reading of papers (when the data extraction form was filled out), since the researchers conducting the study assessed individually their assigned primary studies. This threat was ameliorated by following the pre-defined protocol, carrying out several dry runs individually, and consolidating the differences collaboratively. Though, given that this step involves human judgment, the threat could not be eliminated.

External Validity

External validity concerns establishing the domain to which a study's findings can be generalized. The scope of this SMAP was on reuse through patterns of RE knowledge for the elicitation and specification stages that spanned until 2015. Since it was not attempted to generalize the SMAP conclusions beyond this scope, this validity threat does not apply.

Conclusion Validity

Conclusion validity is concerned about whether the research is reproducible with similar results and whether the conclusions drawn are reasonable given the data.

Repeatability requires detailed reporting of the research process. In this regard, all the steps performed in the systematic mapping were explicitly described by detailing the systematic mapping protocol, as well as the actions taken to reduce possible threats to validity. Repeatability was also aided by the use of existing guidelines.

A threat in interpreting the data is researcher bias. Being researchers in the area of RE and information systems, there was a risk that the researchers conducting the study were biased by their experience and collaborations in the selection and the analysis despite the effort to avoid it. For example, some selected studies of the mapping involve previously the researchers conducting the study or their colleagues. In particular, the papers presenting the PABRE approach (P21), which corresponds to the approach presented in this thesis, includes works of the researchers. This was mitigated by basing the analysis of this specific approach only on the retrieved papers and not on the researchers' knowledge of the approach. In addition to that, although there was a careful analysis of the primary studies, other researchers may find that some aspects may have been neglected in the analysis.

Another threat to validity is related to searching exclusively works in English. Though it is the larger used language in research, there are many active communities in non-English-speaking countries who may propose interesting researches related to the topic.

2.2.5 Conclusions of the Study

This SMAP recalls a great set of publications dealing with patterns to reuse RE knowledge during the elicitation and specification stages. This subsection returns to the main research questions of this study (RQ 1.1.1, RQ 1.1.2, RQ 1.1.3 and RQ 1.1.4) and replies to them according to all the retrieved proposals.

RQ 1.1.1 What is the chronological overview of the research done so far in patterns for reusing RE knowledge?

Patterns for reusing RE knowledge have been a research topic from 1996 to the current days. As shown in Figure 2.2, the number of papers found (i.e. fulfilling the search criteria) is more or less steady from 1997 to 2010, with a peak in 2006 with 7 papers and a bottom with no papers in 2000. Since 2011, the number of papers has been increasing until now (taking into account that probably the number of papers for that year would be larger if it would have been automatically analysed, and not only some venues would have been searched manually).

Some relevant observations follow when looking at the distribution per years of other aspects of the proposals included in the data extraction form:

- **Venue of publication.** Since 2012 (included), 47 publications have been published. 17 of them correspond to workshops publications and 10 of them to journals. This means that the majority of workshops and journals publications included in this SMAP have been published since 2012, which shows a change of focus on the types of venues where authors are targeting to publish.

- **General classification.** The most common type of proposals after 2010 (included) are: formalization/specification patterns (6 proposals), templates (11 proposals), and patterns with context information (21 proposals).
- **Domain.** Most of the proposals which domain corresponds to the type *General (Not Stated)* have been published since 2010 (15 out of 19 proposals). Regarding proposals dealing specifically with security requirements, there is a peak in years 2011 and 2012 (when 6 out of the 15 proposals are published), while the other 9 are scarcely distributed between the rest of the years.
- **Artefact to reuse.** From the 49 proposals published in 2010-2015, 32 of them are dealing with SNL (which corresponds to the 64% of the proposals dealing with SNL). Only 5 out the 13 proposals reusing use cases are published in that period, meaning reusing use cases using patterns is a decreasingly research topic.
- **Notation used to define the artefact to reuse.** Natural language seems to be the main notation language used since 2010 (40 proposals out of 49 published since then are using it, and the total number of proposals using it is 63). For the UML case, 7 out of 11 proposals using it are published after 2010. Regarding *i**, the rate is even bigger: 6 out 7 proposals using *i** are published after 2010. Finally, temporal formal logic is decreasing as years pass: while in the 1996-2009 period (31 proposals), 4 of them are using it, in the 2010-2015 (49 proposals), only 1 is using it.
- **Structure in which the artefact to reuse is contained.** The distribution between artefacts to reuse contained in a bigger structure or them being isolated units of knowledge is evenly distributed between the years: in the period 1996-2009 (31 proposals), 21 proposals consider a bigger structure (68%) and the other 11 consider their artefacts as isolated (32%); in the period 2010-2015 (49 proposals), 34 proposals consider a bigger structure (69%) and the other 15 consider their artefacts as isolated (31%).
- **Relationships.** The rate of proposals considering relationships among their artefacts to reuse is bigger in the period 1996-2009 (17 out 31 proposals, 55%). In the period 2010-2015, the rate is 43% (21 out of 49 proposals).

- **Arrangement.** There is a trend of proposals not specifying how their artefacts should be classified: in the period 2010-2015, 30 out of 49 proposals (61%) are not stating any value for that aspect, while in the period 1996-2009 the rate is 25% (8 out of 31 proposals). Looking at the distribution between repositories and taxonomies of the proposals in these two periods, the distribution does not present big differences.
- **Metamodel for the artefact to reuse.** 12 out of the 17 proposals incorporating a metamodel for the artefact to reuse are published before 2010, and 5 in or after 2010. Comparing the rate of publications in both periods, for the first one it is 42% (12 out of 31), while for the second one it is 10% (5 out of 49), showing a tendency to not incorporate such metamodel in most recent proposals.
- **Method to construct the artefacts to reuse.** There is a slight increase on the rate of papers proposing a method to construct the artefacts to reuse: in the period 1996-2009, 23% of the proposals have such method (7 out of 31); in the period 2010-2015, 28% of the proposals have it (14 out of 49 proposals). That is a good sign, since it is important also to have a method to know how to construct the knowledge to reuse, especially if it is necessary to build a catalogue for each different domain or organization. As for the rate of proposals proposing a partial method, there are not relevant differences between both periods.
- **Method to reuse the artefacts.** The number of proposals incorporating a complete reuse method is barely the same in the period 1996-2009 (13 out of 31 proposals, 42%) and 2010-2015 (17 out of 49 proposals, 35%). In both periods, the rate of partial reuse methods is similar (23% and 21%, respectively).
- **Method to update the catalogue of artefacts to reuse.** Whereas before 2010, 25% proposals have a method (either a complete method or a partial version of it) to update the catalogue (8 out of 31 proposals), after that date the rate decreases down to 14% (7 out of 49 proposals). A decrease in the interest of such a method is disturbing, since maintaining the catalogue up-to-date is as important as constructing and using it (otherwise, it could become obsolete and unusable).
- **Existence of a catalogue of reusable artefacts.** The rate of proposals incorporating a catalogue is equally spread before 2010, and during and after 2010: 25 out of 31 proposals (80%) and 38 out of 49 proposals (77%), respectively.

- **Tools.** The rate of a tool to ease the use of the proposal is steadily distributed between the periods 1996-2009 and 2010-2015: 10 out of 31 proposals in the first case (32%), and 18 out of 49 proposals in the second case (37%).
- **Tests.** There are 12 proposals before 2010 (out of 31 in the same period, 39%) that have been somehow tested. Similarly, in or after 2010, from the 49 proposals published in that period, 19 of them have been tested (38%).
- **Type of research.** There is a slight increase in the number of empirical studies after 2010 (included). From the 31 proposals published in 1996-2009, 1 of them corresponds to experience reports (3%), 11 of them to other empirical studies (35%) and 19 of them to non-empirical studies (62%). For the 2010-2015 period, when 49 proposals have been published, the figures for the types of study are: 3 proposals (6%), 19 proposals (38%), and 27 proposals (55%) for each type, respectively. It is specially worth noting the increase in experience reports, showing that practitioners are every time more interested in research.

RQ 1.1.2 What are the characteristics of the proposed patterns?

To answer this question, the proposals were evaluated according to their structural characteristics, which in the data extraction form correspond to the groups of data items: *Overview*, *Structure* and *Organization*.

Regarding the overview of the proposals:

- **General classification.** There is a higher number of contributions that deal with patterns with context information, followed by templates and domain/NF patterns (33, 21 and 16 proposals, respectively, as shown in Figure 2.5).
- **Scope.** Most of the proposals (59 out of 80) deal with both the elicitation and specification of requirements (Figure 2.6). Writing guidelines patterns and formalization/specification patterns are never used alone for the elicitation of requirements; the most common types of proposals (patterns with context information, templates, and domain/NF patterns) are mostly focused on both the elicitation and specification of requirements (Table 2.7).
- **Domain.** 36 proposals are not focused in a specific domain, so they are considered to be general (Figure 2.7). From the proposals stating a specific domain to reuse

knowledge, 4 could be easily applicable to any other domain. The most targeted specific domains are security requirements (15 proposals) and embedded systems (5 proposals).

- **Type of requirements reused.** 8 proposals do not state the type of requirements being reused, and for the 4 proposals of process patterns this aspect is not applicable (Figure 2.8). For the proposals stating the types being reused, 15 proposals are dealing with 2 types, while 35 proposals are dealing with 3 (i.e. all types of requirements). The main reused type of requirements are FRs (60 proposals), closely followed by NFRs (57 proposals).
- **Artefacts to reuse.** Surveying the different proposals allowed to identify different forms of knowledge representation (Figure 2.9). Some proposals even incorporate more than one form of knowledge to reuse. Sentences in natural language is the prominent form of reused knowledge (50 proposals), followed by use cases (13 proposals), conceptual models (6 proposals), refinement models (5 proposals) and formalization of requirements (4 proposals). Other well-known RE artefacts (such as activity and sequence diagrams) are not appearing in more than 3 proposals. Sentences in natural language are mostly being reused by patterns with context information and templates proposals (Table 2.10).
- **Notation used to define the artefact to reuse.** Some proposals are dealing with more than one type of notation: 25 proposals use 2 types, and 1 proposal 3 types. Natural language gets the largest figure (63 proposals) (Figure 2.10). Other common notations, but not as usual as natural language, are UML (11 proposals), *i** (7 proposals), and temporal formal logic (5 proposals). The other notations are not used by more than 3 of the proposals. Some of the proposals claim they use some kind of language (for instance, some kind of formal language), but they do not specifically state what language is; that happens in 8 proposals.

As for the structure of the proposals:

- **Structure in which the artefact to reuse is contained.** In 56 proposals, the artefact to be reused is part of a bigger structure, while in 24 proposals it is considered as an isolated unit of knowledge (Figure 2.11). For proposals classified as domain/NF patterns, formalization/specification patterns, patterns with context information, and process patterns, usually the artefact to be reused is part of a bigger structure (Table 2.12). In the case of proposals in the templates and writing guidelines patterns categories, the artefacts to be reused in most of the proposals is not surrounded by any extra information.

- **Structure information.** Most of the proposals considering a bigger structure include metadata (45 proposals) and applicability conditions (39 proposals) (Figure 2.12). The other stages information is less common (14 proposals). In the case of parameters, only 22 proposals include them.

Finally, concerning the organization of the proposals:

- **Relationships.** There are 38 proposals that incorporate relationships, most of them presenting different types of relationships (Figure 2.13).
- **Arrangement.** The proposals are presenting either a taxonomy for organizing the artefacts (12 proposals), or a repository with some classification (29 proposals) (Figure 2.14). However, 39 proposals do not state any kind of arrangement.
- **Classification.** From the 41 proposals that state a specific type of arrangement of the artefacts to reuse, some of them present a specific classification to organize the artefacts to reuse. That is the case in 19 proposals (Figure 2.15).

RQ 1.1.3 Which assets are provided with the proposed patterns?

In this question, it was analysed how the proposals address some additional assets related to the artefact to reuse, which in the form for the data extraction correspond to the data items' groups *Metamodel*, *Methods* and *Catalogue*, and to the data item *Tools*.

In regards to metamodel:

- **Metamodel for the artefact to reuse.** Only 17 proposals state to incorporate a metamodel for the artefact to reuse (Figure 2.16).
- **Metamodel for the arrangement.** Only 3 proposals have a metamodel for the arrangement of the artefacts to reuse (Figure 2.17).

On the subject of methods:

- **Method to construct the artefacts to reuse.** 21 proposals incorporate a well-structured method to construct their artefacts to reuse, while 12 proposals present only some guidelines for doing so (Figure 2.18). Most of the proposals do not consider such a method (47 proposals).

- **Method to reuse the artefacts.** This type of method is the most common in the proposals: 30 proposals have a well-structured method to carry the reuse process, and 17 propose at least some guidelines to do it (Figure 2.19). However, 33 proposals do not take that method into account.
- **Method to update the catalogue of artefacts to reuse.** This is the less stablished method in the proposals: 65 proposals do not include it. There are 4 proposals that present a complete method to maintain the catalogue up-to-date, and 11 proposals that incorporate only some guidelines for the update (Figure 2.20). Finally, one proposal do not present neither a method nor guidelines for the catalogue update, but it claims that the update is managed.

As to the catalogue:

- **Existence of a catalogue of artefacts to reuse.** Most of the proposals (63) include some kind of catalogue with ready-to-reuse artefacts (Figure 2.21). There are catalogues of artefacts in all types of proposals (Table 2.13), not existing a specific type with a big difference in the rate of catalogues: for all the types, the rate is comprised between 70% and 87'5%.
- **Domain of the catalogue of artefacts to reuse.** For the majority of the proposals presenting a catalogue, the catalogue is general enough to be used in another environment (37 proposals) (Figure 2.22). On the other hand, for 25 proposals it is specific for a project. For one proposal that includes a catalogue, it was not possible to determine if it is general or specific. Finally, for 17 proposals this aspect does not apply since they do not incorporate any catalogue.
- **State of the catalogue of artefacts to reuse.** For only 5 proposals their catalogue is finished, while for 12 proposals it is still in evolution (Figure 2.23). For most of the proposals incorporating a catalogue, though, it has not been possible to determine what the state of the catalogue is (46 proposals). As happens with the catalogue domain, the state of the catalogue is not applicable for those proposals that do not incorporate any catalogue (17 proposals). Only 4 proposals include a catalogue that is general and finished (Table 2.14).

With reference to tools:

- **Tools.** Only 28 proposals incorporate tools to facilitate the reuse process (Figure 2.24). Most of these tools are built from scratch, while 3 proposals implement them as plugins of commercial tools. The tools are evenly distributed between the different types of proposals (Table 2.15). For all the types, the specific percentage of proposals including tools is 40%-50%, except for domain/NF patterns and writing guidelines patterns, whose percentages are 19% and 29%, respectively.

RQ 1.1.4 What are the most consolidated patterns?

This question is aimed at identifying the most consolidated proposals of patterns for reusing knowledge during RE. With that purpose, the proposals were analysed individually taking into account if they deal or not with the criteria stated in Subsection 2.2.3.3. However, filtering the proposals by all these criteria narrowed the results to nothing. Therefore, it was decided to be less strict with the criteria and select only the ones that are considered more important to facilitate the integration of proposals in real environments. Specifically, the analysis is focused on knowing what proposals are dealing with:

- Both the elicitation and specification of requirements. Applying this filter reduces the set of proposals to 59.
- A general domain (or a domain considered generalizable). This filter leaves the set of proposals in 26.
- All types of requirements (since NTRs are sometimes included in NFRs, for the analysis it is enough if the proposals are dealing with FRs and NFRs). This criterion leaves the set of proposals in 20.
- Methods for constructing and using the artefacts to reuse (it does not matter for filtering if the method is a complete method or just some guidelines, i.e. a partial method). Applying this requirement downsizes the set of proposals to 8.

The 8 proposals considered as the most consolidated ones are: P16, P21, P41, P45, P49, P50, P61, and P66. P21 corresponds to the proposal presented in this thesis; because of that, it is left out of the analysis of this research question. Therefore, the final set of proposals considered as the most consolidated ones is composed by 7 proposals.

All the proposals are focused on reusing SNL corresponding to actual requirements, and in the case of P16 it is also focused on reusing feature models. Regarding the type of proposal, 2 of them correspond to templates (P45, P50), while the rest correspond to patterns with context information.

Table 2.20 shows some details about the proposals. As can be seen, most of the proposals fail to fulfil one aspect or another. Specially, only 3 proposals have a method to update the catalogue, and only 3 have a tool that ease the use of the proposals. In addition, although all the proposals apart from P16 incorporate a catalogue of ready-to-reuse artefacts, they are specific for a project, so probably they are not generalizable.

Table 2.20 – Details of the most consolidated SMAP proposals

Proposal ID	P16	P41	P45	P49	P50	P61	P66
Proposal classification	P	P	T	P	T	P	P
Artefacts to reuse	SNL, FM	SNL	SNL	SNL	SNL	SNL	SNL
Relationships	✓	✓	X	X	X	✓	✓
Classification	X	X	X	✓	✓	✓	✓
Artefact metamodel	X	✓	X	X	X	✓	X
Classification metamodel	X	X	X	X	X	✓	X
Update method	Partial	X	X	Complete	X	Partial	X
Catalogue	✓	✓	✓	X	✓	✓	✓
Catalogue domain	Project	Project	Project	NA	Project	Project	Project
Catalogue state	NS	NS	NS	NA	NS	NS	Finished
Tools	✓	✓	✓	✓	X	X	X
Tests	✓	✓	✓	✓	✓	✓	X

P → Patterns with context information; *T* → Templates; *SNL* → Sentences in natural language; *FM* → Feature models
✓ → Yes; *X* → No; *NA* → Not apply; *NS* → Not stated

Taking into account the details on the table, the most consolidated proposal is P61, from Seungyun et al., that is missing only a complete update method (though it presents some guidelines to do it), and a tool that implements the proposal. As for the catalogue, it incorporates one, but is specific for a project.

2.3 State of the Practice: Requirements Reuse and Patterns

To explore the current state of the practice of requirements reuse and requirement patterns (RPs) based approaches, an empirical study based on an online survey was carried out. The survey does not only investigate the use of RPs based approaches at the practice, but also the perception of participants about the benefits and drawbacks of such approaches. The study was conducted by the applicant and her advisors, and the participants were IT people with industrial experience in RE. The survey was implemented as an Internet questionnaire, for which 71 responses were obtained from requirements engineers with industrial experience in the field.

Although the main focus of the survey is to investigate the current state of the practice of just RPs, it was also noticed while planning it that there were few empirical studies that show the state of the practice of requirements reuse in industry (as shown in Subsection 2.3.1). Most works have only a few questions that address RE. Others try to obtain data about reuse rates and benefits and drawbacks of reuse, but they do not deal with specific requirements reuse techniques. The only existing work that addresses requirements reuse by means of patterns is based only on a few interview answers. Works that give more evidence on the situation, providing observations on the current reuse practices, can benefit both RE practitioners and researchers. Therefore, it was decided that the survey would also explore the current situation of requirements reuse practices in organizations.

Once the results of the survey were analysed, observations were derived about the state of the practice of the respondents' organizations. These observations are related to the three research questions of the study, which correspond to the decomposition of RQ 1.2 (see Subsection 2.3.2 for further information): what is the current state of the practice of requirements reuse in organizations?; what is the opinion of requirements engineers on obstacles to requirements reuse?; and what is the opinion of requirements engineers on the benefits and drawbacks of requirement patterns?.

The chapter is organized as follows. Subsection 2.3.1 presents a background on empirical studies on requirements reuse. Subsection 2.3.2 describes the research questions of the study and the research approach used. In Subsection 2.3.3, the relevant variables that characterize the participants of the survey are stated. Subsections 2.3.4, 2.3.5, and 2.3.6 present the observations obtained regarding each research question, which in some cases are related to the characteristics of Subsection 2.3.3. In Subsection 2.3.7, the observations derived from the survey results are summarized, and, in Subsection 2.3.8, the threats to validity of the survey are described. Finally, Subsection 2.3.9 presents the conclusions.

2.3.1 Background on Requirements Reuse Empirical Studies

This subsection contains the background work on empirical studies that present industrial experiences on requirements reuse, or surveys or interviews that include results about the situation of requirements reuse in organizations. The empirical studies found in the literature that address requirements reuse can be organized in three groups: industrial case studies; surveys and interviews on RE in general; and surveys and interviews on requirements reuse specifically.

Industrial case studies. In 1997, Lam et al. [111] already reported the low percentage of papers on software engineering that include industrial validation. More recently in 2015, the results of the SMAP on reusable knowledge on security requirements of Souag et al. [100] corroborate this fact, since in their study only the 10.5% of the identified papers perform an experimental validation in industry. Therefore, it is not surprising that only seven papers that present industrial case studies to analyse requirements reuse were found (Table 2.21).

Table 2.21 – Related work on industrial case studies on requirements reuse

	Context	Evaluation
Eriksson et al.	Comparison of the PLUSS requirements reuse technique in the context of product lines regarding the copy and paste technique used in past projects of the company under study.	PLUSS performs better than copy and paste for reusing requirements, especially for ROI (in the long term).
Goldin and Berry	Implementation of a requirements reuse initiative in a family of projects in a company until reuse became systematic (which occurred after the 3rd project).	Reuse level is good and the resources used decrease after requirements reuse became systematic.
Issa et al.	Approach used during 18 months in 6 projects.	Time saved in requirements engineering phase.
Karatas et al.	Software engineers checked SRSs for two projects derived from a requirements repository.	Improvement of the quality of SRSs.
Pacheco et al.	Two teams did the specification of requirements, one that just was allowed to query past project SRSs and the other with the catalogue and using the authors' proposal.	Time saved in RE phase; Improvement of the quality of requirements specifications.
Rine and Nada	Demonstrating the effectiveness of reuse not just for requirements, but also for domain models and even code, after 27 case studies in product lines.	Reuse (not just of requirements) decreases development effort, development time and time-to-market, and increases product quality.
Toval et al.	Development of the SIREN repository, which helps to make security issues explicit from the early steps of a system development process.	Time saved in requirements engineering phase.

Three of the seven studies focused on different requirements reuse techniques not related to patterns (Eriksson et al. [50], Goldin and Berry [57], and Rine and Nada [112]). The conclusions of the work reported by Eriksson et al. are that the requirements reuse technique under study (PLUSS) performs better than copy and paste for all of the variables considered. In the specific case of ROI, the authors conclude that, although immediacy gives an advantage to the copy and paste technique, the sustainability of requirements specifications in the long term gives to the PLUSS technique an advantage. In the case of Goldin and Berry, the conclusions are positive for the reuse level and also for the decrease in resources used. Two and a half years after the collaboration, four people were interviewed and confirmed that the process improvements that began with the third project of the case study are sustained and that the savings in resources due to reuse continue, although they could not quantify them. The conclusions of Rine and Nada are in the same line of the previous works, stating that reuse decreases the development effort (or increases productivity), increases the level of product quality, decreases the development time, and decreases the time-to-market.

The other four of the seven studies focused on analysing pattern-based requirements reuse techniques. These studies were extracted from the SMAP presented in Section 2.2. Considering the 80 proposals of the SMAP, 31 include some type of tests of their proposals, but only 11 conduct the study in industry. Furthermore, in only 4 of the 11 proposals, the study consists in the industrial application of the proposed requirements reuse artefacts and in the measurements of the benefits and drawbacks of the reuse. These proposals are the ones of Issa et al. (P31, [113]), Karatas et al. (P37, [114]), Pacheco et al. (P47, [115]) and Toval et al. (P72, [116]).

All of the studies in the before-stated proposals show benefits in using requirement patterns, being the aspects that they measure the time required for specifying requirements, the quality of the resulting specifications and the completeness of the repository of knowledge to reuse considering the number of requirements reused. Three of them validate that the time required for the RE phase decreases because of the use of patterns (Issa et al., Pacheco et al., and Toval et al.). Two of them also report an improvement in the quality of the resulting SRSs (Karatas et al., and Pacheco et al.) and other three make observations about the repository of patterns to reuse and the amount of requirements reused (Issa et al., Karatas et al., and Toval et al.).

Surveys and interviews on RE in general. Several surveys and interviews have been conducted about RE in general that do not address reuse, nor do the concept appear in any of their answers (e.g. the studies of Sadraei et al. [39], Solemon et al. [41], Hall et al. [117], Neill and Laplante [118], and Verner et al. [119]). Below, the ones that include one or more specific questions on requirements reuse are reported. Table 2.22 is dedicated to the seven studies including some questions about requirements reuse. Most of them obtain data only from companies in a specific country, and not globally. Another fact to note is that most of the studies allow participants with different roles, and not only people in charge of eliciting and specifying requirements. The results are that among 13 and 82% of the participants, depending on the study, state that requirements reuse is a practice always or widely followed in their organizations. In the following, some of the results of the studies are highlighted.

Table 2.22 – Related work on surveys and interviews on RE in general that somehow address requirements reuse

	Type of Study	# People	# Organizations	Country	Participants that stated to reuse requirements	Further remarks
Cox et al. [40]	Interview	10 RE experts	10	Australia	40%	Requirements reuse is difficult unless the domain context of new projects is the same as the one of previous projects.
Iqbal et al. [120]	Survey	108 Different roles	18	Malaysia	82%	---
Khankaew and Riddle [121]	Interview	10 Different roles	11	Thailand	75%	---
Matulevicius [122]	Survey	28 Different roles	28	Lithuania	50%	---
Niazi et al. [123]	Survey	39 Req. Engineers	39	Global	61%	64% participants think that requirements reuse can lead to high or medium perceived benefits.
Nikula et al. [124]	Interview	15 Different roles	12	Finland	13%	8 participants use templates or checklists related to requirements.
Méndez and Wagner [125]	Survey	58 Different roles	Not Stated	Germany	---	The use of document or artefact templates is considered positive.
Solemon et al. [126]	Survey	64	Not Stated	Malaysia	97%[a]	---
Tahir and Ahmad [127]	Survey / Interview	27 / 5 Different roles	25	Malaysia	77%	---

[a] Includes participants that state the practice as occasionally used.

Cox et al. [40] interview ten RE experts to analyse the perceived value of Sommerville and Sawyer's RE practices [128] [129], one of them being the reuse of requirements. They conclude that the perceived value of requirements reuse for companies engaged in new development projects is low; these companies considered that it is difficult to reuse requirements unless the domain context of new projects is the same as the domain context of previous projects. However, in the case of interviews with companies that are involved in product line projects, the perceived value is fundamentally important, since product line projects belong to a single context and requirements that define features in the domain can be reapplied in later products and/or releases. Méndez and Wagner [125] collect opinions on the use of RE standards. Even though they do not include any question on requirements reuse explicitly, they conclude that requirements engineers considered the use of document or artefact templates to be positive. Niazi et al. [123] also propose Sommerville and Sawyer's RE practices to 39 experts in global software development projects to determine in which of them they perceive higher benefits. The results show that 25 experts think that requirements reuse can lead to high or medium benefits. Nikula et al. [124] report that 15 interviewed RE experts working for small and medium companies in Finland use templates or checklists to support requirements reuse. In the same study, the need to reuse both requirements and domain knowledge is also identified and perceived as necessary by almost 50% of the participants.

Surveys and interviews fully focused on requirements reuse. Table 2.23 is dedicated to the three publications identified that present surveys or interviews focused on reuse during RE activities (only data and opinions that are relevant for the study presented in this section are included).

Bakar and Kasirun [130] conduct a survey on requirements reuse in Malaysian software development, IT consultancy, research, and education companies. They obtained 36 answers. The majority of participants reuse requirements in an ad-hoc manner and just 19.4% are involved in systematic reuse processes. They report benefits in the reuse of requirements, but they state as barriers of reuse the low quality and incompleteness of requirements that are available for reuse. Although the survey includes questions about the implication of project team members and project management when requirement reuse is applied, the authors do neither report nor analyse the answers to these questions.

Chernak [53] analyses the state of the practice and benefits of reuse in general by means of an online survey. The survey is based on 82 valid responses of participants contacted through professional e-mail groups, IT-related websites, and direct e-mail. One of the most relevant results of this survey is that, although only 49% of participants state having adopted requirements reuse, 93% of them believe that reusing requirements is important and can provide benefits. Other results indicate that the respondents also think that the way of implementing reuse in practice is not clear and that one of the main obstacles might be related to the maintenance and difficulty in using the requirements repository and the low quality of reusable knowledge that companies have. From the answers, it has also been observed that the level of adoption of reuse differs depending on the size of the project team. Finally, it can be noted that 83% of the participants that reused requirements in their last projects (40 of 82 participants), reused requirements in developing new releases of the same application.

Table 2.23 – Related work on surveys and interviews on requirements reuse

	Type of Study	# People	# Organizations	Country	Further remarks
Bakar and Kasirun	Survey	36 Different roles	---	Malaysia	**Requirements reuse.** 72.2% participants apply requirements reuse = 19.4% systematic reuse + 52.8% ad-hoc reuse. **Benefits.** Requirements to be reused are easy to understand as compared to defining new requirements; Reuse gives positive impact to the RE performance; Reuse increases the productivity of the development team. **Why not reuse requirements.** Requirements from previous projects are incomplete or do not exist; Existing requirements are poorly structured; Existing requirements are not kept updated. **Critical factors.** Existence of a tool that facilitates the search and selection of requirements to reuse.
Chernak	Survey	82 50% Business Analysist 50% Different roles	82	Global	**Requirements reuse.** 59% use reuse in their latest projects; The level of adoption of reuse differs depending on the size of the project team. **Benefits.** Faster time-to-market; Lower development cost. **Why not reuse requirements.** Low maintenance of the reuse repository; Requirements to be reused are incomplete; Difficulty of identifying requirements to reuse. **Critical factors.** The way of implementing reuse is not clear; Reusable knowledge was of low quality.
Hoffmann et al.	Interview	5 Requirements Analysts	5	---	**Requirements reuse.** Convenient for recurring requirements (non-functional requirements and recurrent functional ones). **Benefits expected.** Efficiency in elicitation; Understandability of requirements; Completeness of SRSs; Requirements quality; Comparability of requirements; Traceability. **Critical factors and barriers.** Adaptation of the strategy of introduction to each organization; Patterns with suitable language and detail; Organizational changes accepted by all people implied; Provision of information and support to pattern users; Guiding users using requirement patterns; Well- defined reuse method; Clarity in who is responsible of patterns maintenance; Existence of tool support.

Hoffmann et al. [55] interview 5 requirements analysts to know their opinion on the advantages and drawbacks of requirements reuse through patterns. The five analysts had not used RPs before the interviews. In general, they foresee advantages in employing RP approaches within an organization. Four of them foresee efficiency in elicitation, three of them understandability and completeness of SRSs and two of them better requirements quality, comparability and traceability. Specifically, they consider that it is advisable to define RPs for NFRs and recurrent FRs. Finally, among factors and barriers that are relevant for the success of practical application of patterns, they highlight the following: the existence of a well-defined reuse method and its application process (indicating the specific way of introducing it for each organization); the definition of the RPs in an appropriate language and detail level; the acceptance and implication of requirements engineers and managers on the reuse process; the guide and support to RPs users; the clarity of who is responsible of adding and maintaining patterns and the existence of tool support.

2.3.2 Research Approach

Goal. The goal was to conduct an exploratory study of the practices in requirements reuse that are currently being used in organizations and to study in more depth the possible benefits and drawbacks of the use of RPs as a requirements reuse technique.

Research questions. The research question that drove this study, which has been already presented in Section 2.1, was decomposed in three subquestions (Table 2.24). In the following, these subquestions are further explained:

- *RQ 1.2.1: What is the current state of the practice of requirements reuse?* Here it was investigated the current situation of requirements reuse practices in organizations (i.e. the level of requirements reuse, the type of requirements that are more likely to be reused, and the techniques used to implement the concept).
- *RQ 1.2.2: Why are existing requirements reuse proposals not being used in industrial practices?* Taking into account the evidence gathered in the analysis of related work (see Subsection 2.3.1), it was interesting to explore the reasons that hamper the adoption of requirements reuse in organizations and to report all of the identifiable barriers for such adoption.
- *RQ 1.2.3: What benefits and drawbacks can emerge from the use of a catalogue of RPs?* Based on the evidence that patterns are being increasingly proposed by the

scientific community as a means for implementing reuse, it was interesting to ask participants their opinion about whether (and how) RE problems could be mitigated by the existence of an RP catalogue and about critical aspects and barriers for its introduction in an organization.

Table 2.24 – Research question 1.2 and its subquestions

RQ 1.2	In the field of requirements engineering, what is the state of the practice of requirements reuse techniques in industry, and specifically of the techniques based on patterns?
RQ 1.2.1	What is the current state of the practice of requirements reuse?
RQ 1.2.2	Why are existing requirements reuse proposals not being used in industrial practices?
RQ 1.2.3	What benefits and drawbacks can emerge from the use of a catalogue of RPs?

Research method. In order to achieve the goal, different empirical research methods could have been used: survey questionnaires, survey interviews, data aggregation of evidence from industrial case studies, etc. Even though interviews have clear advantages over questionnaires (since an interviewer can clarify doubts about questions and it is possible to extract data in more detail), they have two clear disadvantages: the time required to collect the same number of answers and the less diverse population that it is possible to reach. Therefore, it was decided to use an exploratory survey questionnaire (more specifically an Internet questionnaire) because, as stated by Dillman et al. [131], they allow to collect more data and obtain answers from a wider scope of countries, companies, project types, etc. Considering that questionnaires of this kind are usually rigid and that the choices proposed in the answers of questions may influence the results obtained, the questionnaire was designed in a manner that minimized its influence on the results.

Survey design. As Wohlin et al. states [108], surveys collect qualitative and quantitative information to provide a "snapshot" of the current status related to a phenomenon. To ensure rigor and repeatability and to reduce researcher bias, the survey protocol was designed following the template proposed for evidence-based software engineering[9]. It included 33 questions structured into 8 sections that may be publicly accessed[10]. The questions were chosen with the goal of covering the three subresearch questions (see the relationship between survey sections and RQs in Table 2.25). It was tried to cover all the possible answers in multiple-answer questions so as to not influence the results, always allowing the respondent to state alternative choices that were not explicitly offered. In order to cover the most frequent possible answers, the options were

[9] http://community.dur.ac.uk/ebse/resources/templates/SurveyTemplate.pdf
[10] http://www.upc.edu/gessi/PABRE/SurveyQuestionsThesis.pdf

extracted from the main books and publications on RE (e.g. Hull's et al. [1] and Pohl and Rupp's [6] books).

All the questions on the survey referred either to the RE experience that participants had or to their beliefs according to their RE experience. Sections 2, 3, 4, and 5 of the survey had questions that were related to the participants' background or to general RE practices used in the participants' daily work (such as techniques used or problems faced during elicitation). These questions allowed the sample of respondents to be characterized and allowed to make interesting observations when they were related to the answers of the questions in Sections 6, 7, and 8, which were directly related to requirements reuse and RPs.

Table 2.25 – Relation between survey sections and subresearch questions

Section		Topic	Relation to RQs
Id	Title		
1	**Welcome Page**	Explains the purpose of the survey, who will analyse the results and how they will be communicated	----
2	**Context and Work Experience**	Gathers personal and experience data relevant to the analysis of the survey results	RQ 1.2.1, RQ 1.2.2, RQ 1.2.3
3	**RE Practices**	Surveys general aspects of RE practices followed by the participants	RQ 1.2.1, RQ 1.2.2, RQ 1.2.3
4	**RE Problems**	Includes questions related to the RE problems encountered by the participants in their professional work	RQ 1.2.1, RQ 1.2.2, RQ 1.2.3
5	**Observations on Requirements**	Presents questions about difficulties found in some specific types of requirements derived from the ISO/IEC-25010 quality standard [132]	RQ 1.2.1, RQ 1.2.2, RQ 1.2.3
6	**Reuse during RE**	Elicits current practices of participants on requirements reuse	RQ 1.2.1
7	**Reuse through Patterns**	Surveys participants' opinion about the benefits and barriers of using patterns as a requirements reuse technique	RQ 1.2.3
8	**Barriers to Reuse Adoption**	Explores the participants' opinion on the failure to implement reuse practices in RE	RQ 1.2.2

Protocol. The survey questionnaire was piloted at REFSQ 2013 (April 2013), where it was presented as part of the Empirical Track[11]. The conference attendees were encouraged to respond during the conference. As a consequence of this experience, some changes were implemented in Section 5 of the questionnaire, where a high percentage of the non-completed attempts occurred; these were not changes in the survey questions but changes in the interface to the user. For instance, there was a table at the start of Section 5 containing the definitions of the different types of requirements; since a lot of non-completed attempts occurred in this section, that table was deleted and these definitions were incorporated as tooltip texts wherever a specific type of requirement was mentioned, thereby reducing the percentage of non-completed responses. The questionnaire was available from April 2013 to July 2014.

[11] http://refsq.org/2013/empirical-track/

Channel. The survey was implemented using LimeSurvey[12], which offers support to develop Internet questionnaires and collecting and managing their data.

Population, sampling frame, and sample. The theoretical population for the survey were IT professionals with industrial experience in RE. Finding a suitable sampling frame (i.e. the actual population) is very difficult in surveys for which no exhaustive register of the target population exists [131]. Thanks to the Internet, it has been possible to have access to groups who would have been difficult, if not impossible, to reach through other channels [133]. These groups were the following: requirements engineers that belong to LinkedIn RE groups (namely Requirements Engineering, Requirements Engineering Specialists, Reuse of Requirements Engineering, Requirements Management and Analysis); attendees at RE-related conferences where some publicity was done through papers, posters, and demos (REFSQ 2013, RE 2013, REFSQ 2014); attendees of tutorials at international conferences (RCIS 2013, ICSE 2013, ICSE 2014) and seminars at universities (UFES, Brasil, July 2013; U. Oulu, Finland, March 2014) which were taught by one of the conductors of the study; and finally readers of online RE magazines and communities (the IREB magazine, the Spanish Software Quality Community). The population was proactively reached out to boost participation: new discussions were started every once in a while to disseminate the survey on LinkedIn groups, as recommended by Dillman et al. [131]; reminder messages to tutorial attendees were sent after the conferences; and e-mails were resent to certain RE communities.

In order to avoid bias, and keeping in mind that the composition of the surveying frame could include people that did not fit the population, a question was included in the first section of the survey to identify the answers of the population that was of interest for the study. Finally, 77 valid responses were obtained from 142 respondents who started to answer the survey. Of the 77 valid responses, 71 of them belong to the population that was of interest for the study, and the other 6 respondents were RE researchers with no industrial experience in RE. The potential number of requirements engineers that the survey announcement could reach was more than 30,000. However, the real number of people that really read it remains unknown because it was not possible to know how many people read a post in a LinkedIn group.

[12] http://www.limesurvey.org/

Data analysis. To ensure the quality of the data obtained from the questionnaire, sanity checks were applied to find obvious errors in data. Descriptive statistics were used to analyse the data [134]. Content analysis [135] of the free text answers was performed: these answers were coded into categories, classified, and their frequencies were analysed. Finally, cluster analyses [136] were carried out to find relationships among results, running these tests over each pair of questions made in the survey. The results presented here show only those correlations and cluster analyses that provided significant conclusions.

2.3.3 Characterization of the Respondents

In this subsection, the 71 responses of the survey are described regarding certain aspects related to the respondents' background and experience.

Industrial experience as requirements engineer. Figure 2.27 shows the distribution of the 71 participants according to their level of industrial experience in RE together with their affiliation as industry or academy professionals. The majority of them are industry professionals. However, there are other answers from academic researchers who declare some degree of industrial experience in RE. It is worth noting that participants with significant industrial experience in RE represent more than 75% of the respondents.

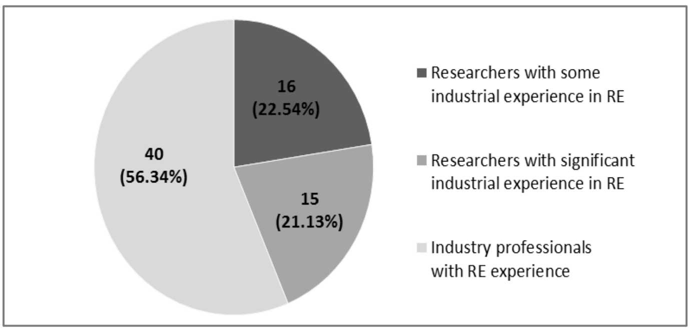


Figure 2.27 – Distribution of responses for the level of industrial experience related to RE

Worldwide distribution. Since the survey was conducted online, requirements engineers from all over the world were able to participate. Figure 2.28 shows the allocation of the participants on the world map, which shows that responses come from 27 countries from all continents, with a special focus on Europe (31 participants; 43.66%) and North America (18 participants; 25.35%).

Figure 2.28 – Global view of survey participant locations

Educational background. Figure 2.29 shows the distribution of participants by their educational background. It is worth noting that more than 75% of the participants have an *MSc* or even a *PhD in Science*. Of the 6 participants that select the option *Other*, 4 have education in business analysis, while the other 2 do not explicitly state their level of studies.

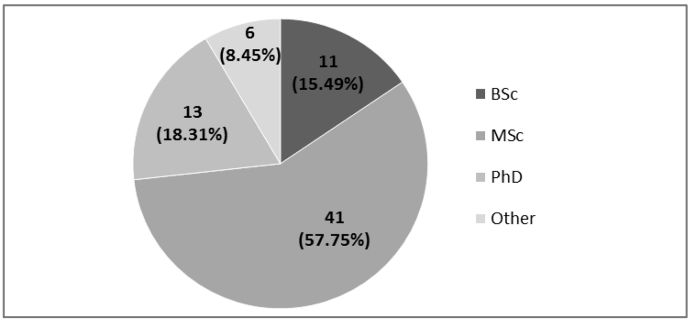


Figure 2.29 – Distribution of responses for educational background

Years of experience. As Figure 2.30 shows, most of the survey participants have more than 5 years of experience (62; 87.32%); specifically, 31 of them (43.66%) have more than 15 years of experience.

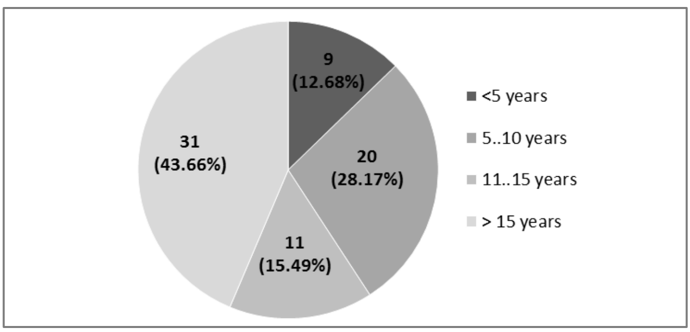


Figure 2.30 – Distribution of responses for years of experience

Organization size. Figure 2.31 contains the distribution of the respondents according to the size of the organizations in which they acquired industrial experience related to RE. The participants were allowed to select more than one size of organization (i.e. it was a multiple answer question). As can be observed, the figures in the categories are quite similar (except for the companies with less than 10 employees), assuring a good coverage of all the possible organization sizes.

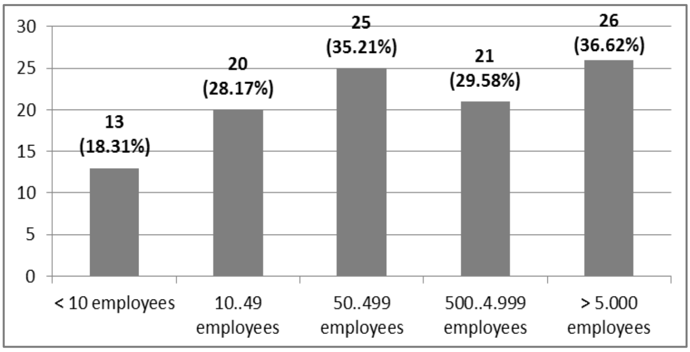


Figure 2.31 – Distribution of responses for organization size (multiple-answer question)

Table 2.26 – Distribution of responses for project domain (multiple-answer question)

Domain	# Respondents	Percentage
Consulting	25	35.21 %
IT Provider	18	25.35 %
Telecommunication	16	22.54 %
Embedded Systems	13	18.31 %
Manufacturing	9	12.68 %
Education	8	11.27 %
Healthcare	7	9.86 %
Insurance	7	9.86 %
Public Administration	7	9.86 %
Transportation	7	9.86 %
E-commerce	6	8.45 %
Finance	6	8.45 %
Automotive	5	7.04 %
Customer Relationship Management	4	5.63 %
Travel	2	2.82 %
Power Distribution	1	1.41 %
Human Resources	0	0.00 %

Organization sector. The sectors of the organizations in which participants acquired industrial experience related to RE (Table 2.28) were presented as a multiple-answer question. The initial list of domains was based on Neill and Laplante's work [118] and refined based on the experience of the conductors of the study. The most common sectors are: *Consulting* (25;

35.21%), *IT Provider* (18; 25.35%), *Telecommunication* (16; 22.54%), and *Embedded Systems* (13; 18.31%). All the other sectors (except *Human Resources*) are represented in the survey but are selected by less than 10 participants.

Languages to specify requirements. A multiple-answer question was used to determine the languages used to specify requirements (Figure 2.32). The main source of choices offered to the participants was extracted from Pohl's RE book [13] and, based on the experience and knowledge of the conductors of this study, those values that were considered to be the most common ones were selected. According to the results, the largest share of responses uses *Natural Language* (57; 80.28%), closely followed by *Use Cases or other scenario-based approaches* (55; 77.46%), and *UML* (38; 53.52%). It is important to note that of the 12 respondents (16.90%) that select the option *Other*, 4 use BPMN to write requirements.

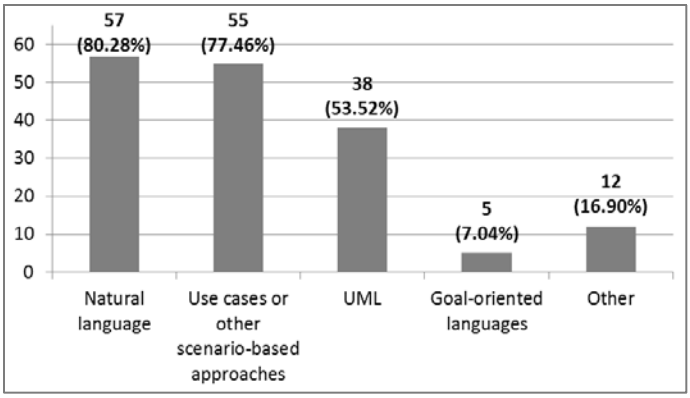


Figure 2.32 – Distribution of responses for languages used to specify requirements (multiple-answer question)

Requirements elicitation methods. A multiple-answer question was used to determine the methods used to elicit requirements (Figure 2.33). The elicitation techniques presented in Hull's et al. [1] and Pohl and Rupp's [6] RE books were used as sources for the proposed answers. Again, based on the conductors' experience, the most common ones were selected for the list presented to the user. The results show that 59 participants (83.10%) use *Interviews*, 50 use *Workshops* (70.42%), 38 use *Questionnaires* (53.52%), 37 use *Observations* (52.11%), 29 use *Focus Groups* (40.85%), and 22 use *Perspective-Based Reading* (30.99%). Other elicitation methods used by 10 of the participants (14.08%) include business-form analysis, prototyping, and their own patented methods. Four participants (5.63%) *Never or rarely use an elicitation method.*

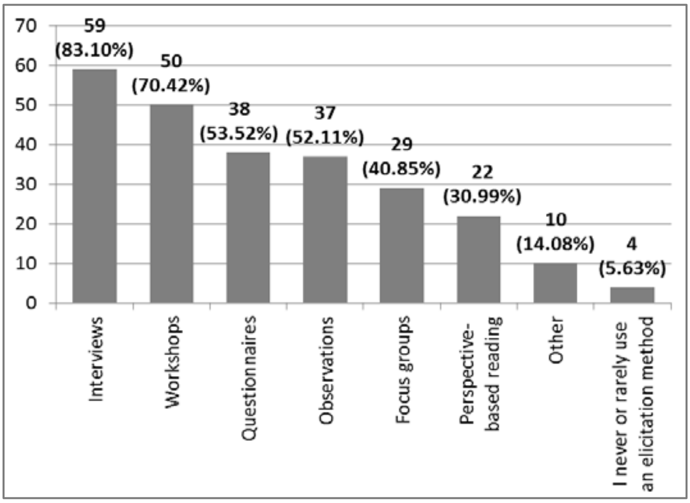


Figure 2.33 – Distribution of responses for requirements elicitation methods (multiple-answer question)

2.3.4 On Requirements Reuse Adoption

In order to answer RQ 1.2.1, participants were asked about three different aspects (see Sections 5 and 6 of the questionnaire): the level of requirements reuse they had in their projects, the techniques they implemented to achieve requirements reuse, and the types of requirements that were more similar among their projects.

Current state of requirements reuse. Participants were asked to measure the level of requirements reuse in their projects using a Likert Scale ranging from *1-Inexistent or Very Low* to *5-Very High*. The results show (Figure 2.34) that a majority of participants (78.87%) state some kind of requirements reuse (i.e. the level is declared as equal to or greater than *2-Low*). However, reuse does not seem to be an established practice in IT projects since only 18 of the participants (25.35%) mark it as equal to or greater than *4-High*.

Table 2.27 contains the cross-tabulation between the requirements reuse level stated by the participants and their organization size. Most of the Chi-Square exact tests led to p-values that were smaller than 0.05, which means that there was a statistically significant relationship between these two variables. Specifically, the results in Table 2.27 show that there is a trend towards a higher level of requirements reuse level the larger the organization is. It is worth noting that this correlation was not due to the domain of the projects carried out by the organizations since these sectors did not show any correlation with the organization size or with the requirements reuse level (i.e. a size or level represented a variety of sectors and not a single one).

No other significant relationship was found between any other variable and the requirements reuse level except the one related to requirements reuse techniques (see the next point in this subsection). Remarkably, no other relationship that one might think would exist appeared in the analysis (e.g. between requirements reuse level and years of experience).

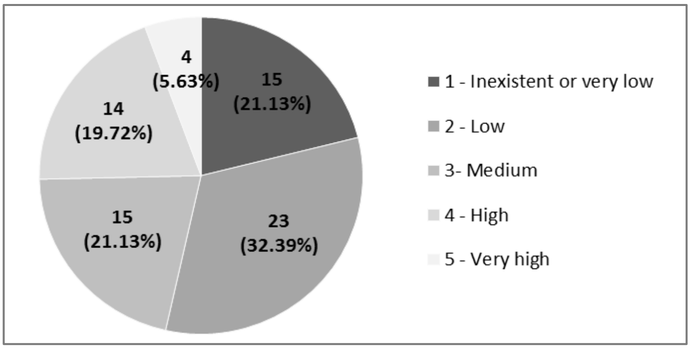


Figure 2.34 – Distribution of responses for requirements reuse level

Table 2.27 – Cross-tabulation of requirements reuse level and organization size

		Organization Size (#employees)				
		<10	10..49	50..499	500..4.999	>5.000
Reqs. Reuse Level	Inexistent or very low	7 (9.86%)	7 (9.86%)	4 (5.63%)	0 (0.00%)	4 (5.63%)
	Low	4 (5.63%)	8 (11.27%)	14 (19.72%)	8 (11.27%)	5 (7.04%)
	Medium	2 (2.82%)	3 (4.23%)	3 (4.23%)	9 (12.68%)	5 (7.04%)
	High	0 (0.00%)	2 (2.82%)	3 (4.23%)	1 (1.41%)	10 (14.08%)
	Very high	0 (0.00%)	0 (0.00%)	1 (1.41%)	3 (4.23%)	2 (2.82%)
TOTAL		**13 (18.31%)**	**20 (28.17%)**	**25 (35.21%)**	**21 (29.58%)**	**26 (36.62%)**
Chi-Square (p-value)		**0.015**	**0.169**	**0.041**	**0.001**	**0.033**

Requirements reuse techniques. Using a multiple-answer question, the participants were provided with a list of requirements reuse techniques. As usual, they had the possibility to add any missing value with an open field value option. This question was only asked to those participants that implemented some kind of requirements reuse in their projects, i.e. those participants stating the requirements reuse level in the first question as being equal to or greater than *2-Low* (56 participants; 78.87%).

The results for this question are shown in Figure 2.35. The most common techniques are those based on the textual copy and subsequent modification (also known as *clone and own reuse*) of requirements from previous projects (used at least by 30 participants; 53.57%). Specifically, these techniques are: *Copy and paste of groups of requirements*, *Copy and paste of individual requirements*, and *Duplicate of a full requirements specification* and work in its parts as needed. Less common techniques are *Fill in predefined templates* and the *Use of a requirement patterns catalogue*; this last technique is the least used (only 6 participants; 10.71%). For 20 participants (35.71%) the reuse technique used is different depending on the project.

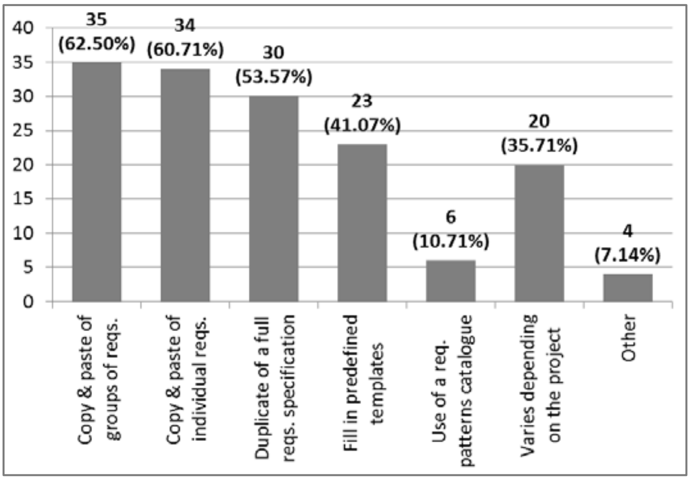


Figure 2.35 – Distribution of responses for requirements reuse techniques (multiple-answer question)

Table 2.28 – Cross-tabulation of requirements reuse level and requirements reuse techniques

		Requirements Reuse Techniques						
		Copy & Paste of groups of requirements	Copy & Paste of individual requirements	Duplicate of a full reqs. specification	Fill in predefined templates	Use of a reqs. patterns catalogue	Varies depending on the project	Other
Reqs. Reuse Level	Low	14 (25.00%)	16 (28.57%)	13 (23.21%)	2 (3.57%)	0 (0.00%)	11 (19.64%)	1 (1.79%)
	Medium	11 (19.64%)	7 (12.50%)	10 (17.86%)	6 (10.71%)	1 (1.79%)	5 (8.93%)	0 (0.00%)
	High	8 (14.29%)	8 (14.29%)	5 (8.93%)	12 (21.43%)	3 (5.36%)	3 (5.36%)	1 (1.79%)
	Very high	2 (3.57%)	3 (5.36%)	1 (1.79%)	3 (5.36%)	2 (3.57%)	1 (1.79%)	2 (3.57%)
TOTAL		**35 (62.50%)**	**34 (60.71%)**	**29 (51.79%)**	**23 (41.07%)**	**6 (10.71%)**	**20 (35.71%)**	**4 (7.14%)**
Chi-Square (p-value)		**0.029**	**0.015**	**0.008**	**0.014**	**0.34**	**0.011**	**0.57**

An in-depth look into the relationship between the reuse level and the reuse techniques (Table 2.28) shows that there is a statistically significant relationship between these two variables (using Chi-Square exact test). The techniques for which the relationship could not be stated are *Use of a requirement patterns catalogue* and *Other*. The reason is that there were not enough data points for these techniques to make reliable the results of the tests. Even taking this into account, it is possible to observe the trend that the more elaborated reuse techniques are (*Fill-in predefined templates*, *Use of a requirement patterns catalogue*), the higher the reuse level is, whereas simpler reuse techniques (*Copy and paste of groups of requirements*, *Copy and paste of individual requirements*, *Duplicate of a full requirements specification*) are mostly used in organizations with low and medium reuse levels. Finally, it is important to note that respondents participating in projects with lower reuse level declare that they use different requirements reuse techniques depending on the project, indicating reuse techniques seem to be more established in higher reuse levels.

From all the other correlation analyses that were carried out over the results, there is another interesting fact related to the languages that are used to specify requirements and the techniques that are used to reuse requirements. The results highlight a strong statistical relationship (Chi-Square test p-value = 0.018) among the respondents that use natural language to specify requirements and the duplication of specifications for reusing requirements. This indicates that the respondents that use natural language to specify requirements acquire the duplication of specifications as main reuse technique in more than half of the cases.

Types of requirements likely to be reused. The survey included questions to ask about the similarity between requirements of the same type in different projects (based on the respondents' experience). These questions used a Likert Scale, ranging from *1-Totally Agree* to *5-Totally Disagree*. The requirement types proposed were based on the characteristics of the quality models proposed in the ISO/IEC-25010 standard [132] and in the extended version presented by Carvallo et al. [8] of its predecessor (i.e. the ISO/IEC 9126-1 quality standard [137]). The respondents were allowed to add other requirement types that might be relevant and that were not included in the list provided in the survey. The types ranked with a higher reuse rate are: *Reliability*, *Maintainability*, *Usability*, and *Security* (see Table 2.29). For the other types of requirements, the results do not highlight any significant difference in the level of recurrence, with most of them being around 3 (equivalent to Neutral value). Based on the classification of requirement as FR, NFR and NTR, the requirements that are more likely to be reused are NFRs.

Table 2.29 – Average response for requirement types likely to be more similar between projects (1 – Totally agree, 5 – Totally disagree)

Classification	Requirement Types	Likert Scale Average [13]
NFR	Reliability	1.75
NFR	Maintainability	1.93
NFR	Usability	2.17
NFR	Security	2.35
NFR	Performance Efficiency	2.75
NTR	Business Suitability	2.85
NTR	Project Suitability	2.86
NFR	Compatibility	2.93
NFR	Portability	2.94
FR	Functionality Suitability	2.96
NTR	Product Non-Technical Suitability	3.11
NTR	Supplier Suitability	3.13

2.3.5 On Barriers to the Adoption of Requirements Reuse

In order to answer RQ 1.2.2, the survey collected the opinion of those participants who declared a level of reuse as inexistent or very low (15 participants; 21.13%) about two aspects: the possible problems in requirements reuse proposals that prevent them from adopting reuse in organizations; and what is missing in requirements reuse proposals made by researchers to facilitate the incorporation of reuse in industry.

Problems that hinder the adoption of requirements reuse proposals in organizations. The participants were provided with a list of problems for which they could select one or more options; they also had the possibility to add any missing value with an open value option. Figure 2.36 shows that the common opinion is that *Organizations do not know how to do this incorporation* (14 participants; 93.30%). Three other issues considered relevant for almost half of the respondents are the following: *Even if their incorporation may provide benefits, the initial investment is too high* (8 participants; 53.33%); *Organizations never thought about incorporating requirements reuse proposals* (7 participants; 46.67%); and the opinion that *Organizations consider the incorporation of requirements reuse to be too complex* (7 participants; 46.67%). As a summary, it can be concluded that the reasons are based on the ignorance about reuse elicitation processes and on doubts with regard to its return-on-investment.

[13] See *Face Validity* in Subsection 2.3.8 for a discussion on calculating averages over Likert Scale variables.

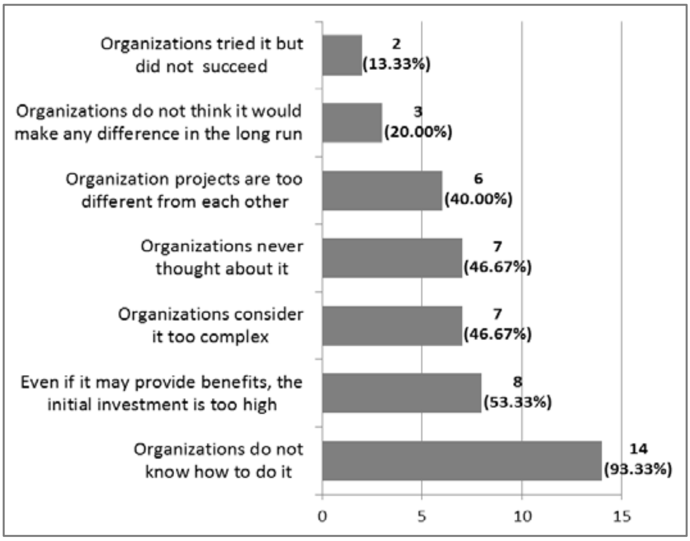


Figure 2.36 – Distribution of responses for problems that hinder the adoption of requirements reuse proposals (multiple-answer question)

What is missing in researchers' requirements reuse proposals to be adopted by practitioners. As the issue was asked with an optional free text question, only 4 answers were collected. Despite this, the respondents made some good points that are worth discussing. On the one hand, two respondents agree on the fact that what is missing in requirements reuse proposals is a solid business case behind them that can convince a CIO to make the investment necessary to incorporate them into the RE process: "*[What is missing is] Presenting successful cases on the existing requirements reuse proposals for new requirements reuse clients.*" and "*A solid business case is needed for requirements reuse to be used. Since there is a lack of solid business cases, another approach that could be used is Technology Maturation. However, there are few companies that are big enough, with deep pockets, and the required imagination to bankroll that approach for requirements reuse.*". On the other hand, the other two respondents declared that the reason for not incorporating requirements reuse in industry is the lack of process maturity in organizations: "*Nothing [is missing], apart from the maturity of the organization.*" and "*With [my] limited knowledge, I do not believe anything is missing, but the maturity of the company may be the reason.*".

2.3.6 On Requirements Reuse through Patterns

To know the benefits and drawbacks of using a catalogue of RPs to elicit requirements (to answer RQ 1.2.3 of the study), in Section 8 of the questionnaire, the 71 participants were provided with a short explanation of what an RP and an RP catalogue are. Afterwards, the participants were asked their opinion about a list of common RE problems that could be mitigated by using an RP catalogue and two lists of critical factors and barriers that could influence its successful adoption. In the three lists, the participants could add new items that might be missing. The values of the first list were extracted by looking at Pohl's RE book [13] and the IEEE 830 standard [138]. To state the values in the second and third lists, several sources were used: the experience of the conductors of the study, the values obtained in Hoffmann's et al. survey addressing requirements reuse [55] and the relevant general requirements reuse barriers and success factors stated by Wiegers and Beatty [26].

Problems mitigated by the use of an RP catalogue. Table 2.30 shows that the four problems that could be most mitigated are the following: *Incompleteness of requirements specification*, *Lack of requirements uniformity*, *Inconsistency of requirements*, and *Ambiguity of requirements*. The main differences among participants not using RPs (65 participants) and the ones using RPs (6 participants) is that the last ones consider that RPs could help them to not *Spend too much time in requirements elicitation*. The respondents added problems that were missing on the list, the most common ones being: *Lack of requirements relationships* (dependencies), *Efficiency of the requirements elicitation process*, and *Accessibility of RE to small and medium sized enterprises*. For those individuals who have used RPs, *The change of stakeholders' needs during the requirements elicitation process* is also a problem likely to be mitigated.

Critical factors for the successful adoption of an RP catalogue. *The existence of a well-defined method for using RPs* as well as *The existence of tool support* are considered to be the most critical factors for the introduction of RPs by all types of respondents (see results in Table 2.31). Remarkably, the respondents who have used RPs give more relevance to *The existence of a community of users supporting RPs*, ranking it in third position. Finally, both groups agree that *The existence of a help desk* is the least significant critical factor.

Other critical factors that were not included in the list but are considered as being very important by the participants are: *The existence of a ready-to-use RP catalogue*, *The existence of a person or department inside the organization expert on RPs*, *The existence of successful cases using RPs*, and *The possibility of having free trial periods*.

Table 2.30 – Average response for problems mitigated by the use of an RP catalogue (1 – A lot, 3 – At all)

Problems mitigated by the use of an RP catalogue	Likert Scale Average [13] (65 participants NOT using RPs)	Likert Scale Average [13] (6 participants using RPs)	Likert Scale Average [13] (all participants)
Incompleteness of requirements specification	1.59	1.66	1.60
Lack of requirements uniformity	1.64	1.66	1.64
Requirements inconsistency	1.76	1.66	1.75
Requirements ambiguity	1.80	1.66	1.79
Lack of requirements quantification	1.86	1.83	1.86
Stakeholders do not know their needs exactly	1.88	1.83	1.88
Too little time invested in requirements elicitation	1.89	2.17	1.91
Requirements non-verifiable	1.90	2.17	1.92
Too much time spent in requirements elicitation	1.93	1.83	1.92
Lack of requirements traceability	1.95	2.33	1.98
Stakeholders' needs change during the requirements elicitation process	2.00	2.33	2.03
Lack of requirements prioritization	2.08	2.33	2.10
Conflicts among needs stated by stakeholders	2.11	2.33	2.13
Lack of requirements relationships (dependencies) [14]	*1.00 (3)*	---	*1.00 (3)*
Efficiency of the requirements elicitation process [14]	*1.00 (5)*	---	*1.00 (5)*
Accessibility of RE to small and medium sized enterprises [14]	*1.00 (3)*	---	*1.00 (3)*
Change of stakeholders' needs during the requirements elicitation process [14]	---	*1.67(3)*	*1.67 (3)*

Table 2.31 – Average response for critical factors influencing the adoption of an RP catalogue (1 – Totally agree, 5 – Totally disagree)

Critical factors influencing the adoption of an RP catalogue	Likert Scale Average [13] (65 participants NOT using RPs)	Likert Scale Average [13] (6 participants using RPs)	Likert Scale Average [13] (all participants)
Well-defined reuse method	1.52	1.33	1.50
Tool support	1.65	1.66	1.65
Training courses	2.09	2.33	2.11
Existence of a community of users	2.22	2.00	2.20
Help desk	2.69	2.50	2.67
Ready-to-use RP catalogue [15]	*1.00 (3)*	---	*1.00 (3)*
Person or department inside the organization expert on RPs [15]	*1.50 (2)*	*1.00 (2)*	*1.25 (4)*
Successful cases using RPs [15]	*1.50 (3)*	---	*1.50 (3)*
Free trial periods [15]	*2.00 (2)*	---	*2.00 (2)*

Barriers for the successful adoption of an RP catalogue. For the respondents not using RPs, only two items from the list of barriers to the adoption of RPs (Table 2.32) are

[14] Further problems stated by participants (in brackets, number of participants that stated them).

[15] Further critical factors stated by participants (in brackets, number of participants that stated them).

considered to be important: *The resistance of requirements engineers to change*, and *The integration of the catalogue with the existing requirements engineering process*. For the respondents that have used RPs, the respondents reinforce the general conclusion that the most important barrier is: *The resistance of requirements engineers to change*. However, differences arise regarding the rest of barriers. *The risk of converting the requirements elicitation into a stiff process* is more important for the respondents that have used RPs (in the second position for these respondents, but on the fourth one for the respondents not using RPs). For *The amount of reusable knowledge to create and maintain*, the participants that have used RPs totally disagree with this being a barrier, as opposed to the rest of respondents, which rank it in the third position with an average of 2.41.

Table 2.32 – Average response for barriers influencing the adoption of an RP catalogue (1 – Totally agree, 5 – Totally disagree)

Barriers influencing the adoption of an RP catalogue	Likert Scale Average [13] (65 participants NOT using RPs)	Likert Scale Average [13] (6 participants using RPs)	Likert Scale Average [13] (all participants)
Resistance to change of requirements engineers	1.92	1.5	1.88
Integration of the catalogue with the existing RE processes	2.03	2.00	2.03
Amount of reusable knowledge necessary to create and maintain	2.41	4.16	2.56
Risk of converting requirements elicitation into a stiff process	2.42	1.83	2.37
Lack of management support [16]	*1.00 (6)*	*1.00 (2)*	*1.00 (8)*
Difficulty of adapting RP output to the organization requirements specification format[16]	*1.00 (2)*	---	*1.00 (2)*

It is important to point out the statistical relationship that exists among the consideration of *The existence of tool support* as a critical factor for adopting RPs and the consideration as barriers of *The integration of the catalogue with existing RE processes* and *The amount of reusable knowledge to create and maintain*. For the first barrier, the Chi-Square test gives a p-value of 0.002 and 41 participants (57.75%) agree on both statements. For the second barrier, the p-value is 0.007 and 34 participants (47.89%) agree on both aspects.

Another statistical relationship found is that among those responses believing that *The existence of a well-defined method* for using RPs is a critical factor for introducing RPs, and that also believe that there is a *Risk of converting requirements elicitation into a stiff process*. The p-value of the Chi-Square test is 0.047 and 40 participants (56.34%) agree on both statements.

[16] Further barriers stated by participants (in brackets, number of participants that stated them).

The participants added other barriers. The most common ones are: *The lack of management support* and *The difficulty of adapting RPs output to the organization requirements specification format*.

2.3.7 Discussion of the Results

This subsection includes the observations, related to each RQ, derived from the analysis of the survey results. A comparison with the empirical works on requirements reuse cited in Subsection 2.3.1 is also presented.

2.3.7.1 About the Current State of the Practice of Requirements Reuse

With regard to level of requirements reuse, requirements reuse techniques and types of requirements more prone to reuse, the following observations were derived.

Requirements reuse is not an established practice in IT organizations. Although 79% of the participants state some level of requirements reuse, only 25% of the participants mark it as equal to or greater than high (see Figure 2.34). In other surveys on RE practices, the average of participants stating the requirements reuse as a practice always or widely followed ranges between 13% and 82% (see Table 2.22). The lower rate (13%) is coming from Nikula's et al. [121] work, probably due to the fact that the participants work in small and medium sized organizations, and also because of the age of their study. In surveys focused on requirements reuse, Chernak's survey results [53] show that 59% of the participants state having used requirements reuse in their last projects. This percentage is lower than in our survey (79%), and is also lower than the one in Bakar and Kasirum's survey [130], where the percentage is 72.2%. It is not possible to compare this aspect to the results of Hoffman's et al. interviews [55] because their study does not include results on the state of the practice but on opinions of requirements engineers.

Participants of larger organizations declare a higher level of reuse. Higher reuse levels in larger organizations (see Table 2.27) could be explained by a higher number of IT projects with similar characteristics in this kind of organizations, which would make that recurrence of requirements in subsequent projects more likely to appear. It is not possible to compare this result with other studies. The only work that explores a similar relationship is the one of Chernak [53], which observes that the level of adoption of reuse differs depending on the size of the project team, but it does not explore the relationship with the size of the organization.

Requirements reuse techniques more commonly used are those based on the textual copy and subsequent modification of requirements from previous projects. The most common techniques (chosen by more than a half of the participants) are those based on the textual copy and subsequent modification of requirements from previous projects (see Figure 2.35). In particular, participants using natural language to specify requirements adopt the duplication of specifications as main reuse technique in more than half of the cases.

This observation shows the existing distance between research and industry in this area, which is corroborated too by the fact that only 5% of proposals on RPs identified in the SMAP are presenting validation as an industrial application of the proposal (Subsection 2.3.1). The surveys on requirements reuse (Bakar and Kasirun [130], Chernak [53]) do not ask about techniques or level of abstraction of the knowledge to reuse, but only about the artefacts or languages used to specify requirements. The reason behind that could be that they assume reuse as a simple copy and modification of knowledge to reuse without considering having levels of abstraction in this knowledge.

There is a correlation between the level of requirements reuse and the requirements reuse techniques used. This study shows that the participants that use more elaborated reuse techniques are the ones that declare a higher reuse level in their projects (see Table 2.28). At the same time, it can be observed the fact that low level of reuse is significantly related to small companies (see Table 2.27). From these facts, it may be inferred that the participants that state a low level of reuse (32%) and less elaborated reuse techniques are probably referring to ad-hoc requirements reuse, i.e. not integrated in the requirements process of the company but as a practice followed by one or more employees or by small companies without consolidated development processes. Higher reuse levels lead to elaborated reuse techniques, since a high level of reuse induces the definition of methods and processes of reuse.

The fact that larger organizations tend to have better-defined, well-known and established methods and processes, which is a critical factor for applying reuse, is corroborated by Dybå [139]. As indicated above, other surveys do not ask about techniques in the same meaning than the one used in this study, and thus it is not possible to check this observation in their results.

Organizations with more established software processes and methods are the ones that declare a higher level of requirements reuse. This was derived from the fact that participants declaring a reuse level low or medium tended to use different requirements reuse techniques in different projects (35.71% of participants). In addition, results show that reuse techniques seem to be more established in projects of participants that declare a reuse level high or very high in their projects. This point is also supported by the answers to questions related to RQ 1.2.2 (Subsection 2.3.5), where the participants declaring an inexistent or very low level of reuse state that the reason is that organizations do not know how to incorporate requirements reuse. In two specific cases, this is supported by the comment about the lack of process maturity in organizations.

This conclusion matches the assumption stated in Somerville's RE maturity model [25] [128] that requirements reuse corresponds to an advanced RE elicitation technique. Other authors (such as Goldin and Berry [57], Rine and Nada [112], Nikula et al. [124], Chernak [53], and Hoffmann et al. [55]) reach the same conclusions about the importance of establishing and adopting well-defined requirements reuse processes. More specifically, in Rine and Nada, it is possible to observe the same trend that organizations with more mature processes reuse more project artefacts (not only requirements, but also models or code): of the 14 organizations that present mature processes, 9 state a high reuse level (64.29%), while of the 13 organizations that have less mature processes, only 5 state high reuse levels (38.46%). After the case study they conducted, Goldin and Berry also state the importance of the maturity of the requirements processes in reuse and the implication of organizations' management.

NFRs are more likely to be similar or recurrent among projects. With respect to the type of requirements that are more prone to be reused among projects, NFRs (rows 1 to 5, 8 and 9 in Table 2.29) are considered as more likely to be reused than FRs (row 10). For NTRs (rows 6, 7, 11 and 12), the results are not the ones expected based on the own experience of the conductors of the current study. For instance, requirements on the *Supplier Suitability*, which are defined in the questionnaire as those requirements that state conditions on the organization that distributes or implements the software product, are considered to be less recurrent than FRs. The interpretation is that NTRs were not well understood by the participants since, according to LIST (one of the collaborators of GESSI), this kind of requirement is in fact quite recurrent. This misunderstanding could be caused by the fact that NTRs are not always included in SRSs unless projects are call-for-tender projects, or because when they are included, they do so as part of NFRs.

Existing works (e.g. the ones of Withall [9], Hoffman et al. [11], and Suppakkul et al. [10], Jaramillo [64], Toval et al. [140] and Shahrokni and Feldt [141]) align with these findings since all of them are software requirements reuse proposals mainly involving requirements that fit the types that are identified as being more likely to be reused (i.e. the ones concerning NFRs). It is also important to remark that, as supported by the previous works too, a big percentage of NFRs and NTRs are domain-independent (i.e. they appear basically in the same way in different SRSs, even if they belong to projects from different domains). For FRs, it is quite the opposite: as stated by Lam [12], it is necessary to identify and formalize the reusable requirements for each functional area. Since FRs are domain-dependent (e.g. Li's et al. work for seismology applications [52], Filipovikj's et al. work for the automotive domain [142], Jensen's et al. work for healthcare applications [143], and Konrad and Cheng's work for embedded systems [144]), it is not surprising that this type of requirement is ranked as less reusable than NFRs. However, in the case of related work addressing reuse in companies working on product lines or on product releases, requirements reuse of FRs is high since they address development of software products in the same domain area, as presented in Cox's et al. [40] and Chernak's [53] studies. Finally, the interview's participants in Hoffmann's et al. survey [55] agree that RPs would be usable for NFRs and recurrent FRs.

2.3.7.2 Reasons behind the Lack of Adoption of Existing Requirements Reuse Proposals

Most of the participants agree that the most common barrier for organizations is their ignorance on incorporating a reuse strategy into their current processes (93.3%). Other barriers considered relevant for almost half of the respondents are (Figure 2.36): the initial investment required, the lack of awareness about the benefits that reuse may bring, and the inherent complexity of implementing reuse. The drawbacks stated by participants in existing requirements reuse proposals are the absence of a solid business case behind the approaches that may convince CIOs and the lack of process maturity in organizations.

Ignorance of reuse techniques and processes is the main reason of the lack of reuse adoption. As a summary, the results show that the main cause for organizations not adopting requirements reuse is that they do not know how to do it, which implies being in ignorance of the techniques and processes behind such reuse. This is corroborated by the main success factors related to the adoption of RPs obtained as answers to RQ 1.2.3 (Subsection 2.3.6). Since these factors and barriers refer to the adoption of reuse and they are not strictly related to RPs, they agree with one of the causes for organizations to not adopt requirements reuse, i.e. *Ignorance of requirements reuse elicitation techniques and processes* (see Figure 2.36). Another implied conclusion of these data is that more empirical research on requirements reuse should be carried out in order to transfer requirements reuse techniques and methods to companies, and to demonstrate the benefits and ROI that they provide (see RQ 1.2.2 results in Subsection 2.3.5).

Bakar and Kasirun [130] and Chernak [53] also ask in their surveys the reasons for not reusing requirements. Both surveys obtain different answers than the one identified in this study, being in their case more based on the knowledge and artefacts to reuse than in the reuse process. The reasons they identify are the lack of quality and incompleteness of requirements to reuse in requirement repositories. In addition, Bakar and Kasirun also identify as critical factors the lack of convenient tools with suitable requirements classification and good facilities for accessing to the requirements repository. The answers of the current survey are different of the surveys previously stated probably because in this survey this question was just answered by participants who declared a low level or no experience on requirements reuse, and in the other surveys the questions were answered by all the participants (having 72.2% of the participants considerable experience in requirements reuse). Probably, the second group stated the problems they are having in requirements reuse applications due to the bad quality of reuse knowledge, and in the current study participants just thought about the doubts about how introducing the practice, the cost of this introduction and the process of applying reuse.

2.3.7.3 About the Benefits and Drawbacks of Using a Catalogue of RPs

The following observations are highlighted with regard to the benefits and drawbacks of the use of an RP catalogue.

Problems mitigated by the use of RPs are mainly related to the quality of the resultant SRSs. The four problems that can be mitigated by the use of RPs (as identified by the survey participants) are related to the quality of requirement specifications (see Table 2.30): *Incompleteness of requirements specifications, Lack of requirements uniformity, Requirements inconsistency,* and *Requirements ambiguity*. This is a logical consequence of working with a knowledge base of reusable artefacts that is supposed to contain artefacts with certain quality. Considering specifically the answers of participants with experience in RPs, one of the mitigated problems is to *spend too much time in requirements elicitation*.

These four quality problems are also identified in the interviews reported by Hoffmann et al. [55]. The interviewees in that study also point out the *Efficiency of the elicitation process* as a benefit from using RPs. This is because they think that less time would be spent on the elicitation process if patterns were used. From the industrial applications of requirements reuse proposals of Subsection 2.3.1, three of them effectively observe a decrease in the time dedicated to RE (Issa et al. [113], Pacheco et al. [115], Toval et al. [116]) and two of them (Karatas et al. 2014 [114], Pacheco et al. [115]) observe an improvement on the requirements quality. The other benefit identified by these interviewees is the *Improvement on the requirements traceability*. In this case, the reason for that benefit is that requirement dependencies would be incorporated in the patterns and propagated to SRSs. In the survey presented in this section, the respondents do not see the relationship of patterns with the efficiency of the elicitation process and the mitigation of the lack of traceability (their Likert scale average was 1.92 and 1.98 respectively, with *1-Agree a lot* and *3-Do not agree at all*). This is especially surprising for the efficiency of the elicitation process because, like all reuse techniques, the use of RPs would be expected to reduce the time invested in the elicitation. It is worth remarking, however, that the participants already using some kind of RP think that the use of this approach could help to spend less time for the elicitation and specification of requirements.

Critical factors and barriers for the successful adoption of an RP catalogue are related to the reuse approach and people involved. The importance given to the existence of a *Reuse method* and *Tool support* (identified as critical success factors in Table 2.31) is probably caused by the absence of a well-defined and mature method to guide the reuse processes undertaken by the participants. It is not surprising that the barriers related to people involvement are considered the most important ones (Table 2.32) since, as stated by Dybå [145], in organizational processes, the involvement of personnel is a key factor for the adoption of a reuse technique and its success.

The critical factors and barriers for the adoption of RPs identified in the current survey are also identified by Hoffman et al. [55]. Other aspects identified by Hoffman et al. related to the quality of the RP catalogue were not in the current survey because the conductors of the current study took them for granted (an RP is not considered reusable if it does not have a good quality).

2.3.8 Threats to Validity

Internet surveys are powerful instruments that make it possible to know the current state of the practice. However, they usually have some weaknesses that threaten their validity, as Evans and Mathur state [146]. In this subsection, the threats to validity of the current study are analysed based on some of the aspects defined by Dillman et al. [131], Wohlin et al. [108], Evans and Mathur [146], and Trochim and Donnelly [147].

Internal Validity

Instrumentation. Instrumentation threats can appear if the survey used for the experiment has an error in its design or changes in the survey are necessary in the middle of the experiment. To avoid this, firstly, pilots of the questionnaire were conducted to ensure its correct understanding and to find possible defects. In addition, a native English speaker revised the questionnaire. As a consequence of these pilots, some changes were implemented in the interface of a specific section of the survey where a high percentage of the non-completed attempts occurred (for more information see Survey Design in Subsection 2.3.2). Secondly, to avoid the typical design errors in Internet surveys, and taking into account the results of the pilots, some critical questions were accompanied with a glossary of terms, and text fields for clarification were added whenever necessary. This glossary of terms included the description of the different NFRs and NTRs types used in the survey as well as the concepts RP and RP catalogue.

Sampling validity. The population of interest for the study was requirements engineers with industrial experience (either pure practitioners or researchers that worked or had worked in industry as requirements engineers). Given their professional scope and skills, it may be assumed that they were an Internet-aware population with enough expertise to answer an online survey without technological impediments.

For the selection of a representative and random sample of this population, social networks and forums where requirements engineers meet each other were the main focus: RE conferences (advertising the survey through two publications [82] [79]), RE LinkedIn groups, and RE magazines (advertising the survey trough a publication [85]). In total, the survey was proposed to approximately 30,000 members through the LinkedIn and community groups. Although the survey was announced mainly to RE practitioners (who were the target population of the survey), the survey had a question to determine the grade of industrial experience in RE in order to be able to filter out responses from IT people with no RE industrial experience.

Finally, several measures were taken to avoid fake answers coming from people trying to sabotage or bias the study. Firstly, in the platform used to implement the survey (Lime Survey), the detection of responses coming from the same IP address was activated, making it impossible for the same person to submit the survey twice without changing the IP address of their device. Secondly, the online questionnaire was organized in sections that were presented on different web pages. One or more questions on each page were impossible to skip so that the work involved in answering the survey would discourage people who were not really interested in the survey subject.

Participants' perception. Since online questionnaires were being used, if the survey proposal was perceived as junk publicity, the outcome of the results could be affected. To avoid this, the survey invitation e-mails were sent to RE conferences attendees and RE communities from the conductors' academic e-mail addresses. The e-mails included a brief text explaining the academic purpose of the survey, its link, and the sign of the conductors' names. A professional image was provided by opening a specific webpage hosting the link (http://www.upc.edu/gessi/PABRE/Survey.html), which was a new tab in the web resource dedicated to the PABRE framework. In the announcements through Internet (especially in the LinkedIn groups), the survey was always presented inside discussion topics and they were maintained alive by participating in the discussions to show potential respondents that the conductors were interested in them answering the survey for academic purposes.

Low response-rate. As stated by Dillman et al. [131], one of the main problems of online surveys (apart from finding a good sampling frame) is having very low participation rates. To avoid this issue, in the case of LinkedIn, a discussion topic with a question to engage people to participate in the discussion was introduced. Then the survey was proposed inside the discussion. This resulted in a higher number of answers compared to just announcing the survey directly inside the groups. For instance, a discussion that gave additional answers was introduced as: *Are you using some requirements reuse practice during requirements engineering?*. As Lethbridge explains [148], even though these low participation rates (which are common in online questionnaires) cannot be used for a rigorous statistical analysis, they can be used to understand trends. Other surveys that address similar contexts and use similar channels have similar or lower participation rates (e.g. Milewski's [149], Umarji and Sim's [150], and Solinski and Petersen's [151] surveys).

External and Construct Validity

Results generalization. Although the participants of the questionnaire were tried to be selected in a random way, the nature of their companies, research centres, and projects was very different. This diversity (added to the fact of having only 71 responses) does not allow for a generalization of the results; only observations about the current state of the practice can be made. Therefore, it is not possible to guarantee with good level of confidence that the conclusions of the analysis correspond to results that would be obtained by conducting the same survey with the entire RE population. Additionally, it is worth pointing out that, in the questions related to RQ 1.2.3 (which is related to the benefits and drawbacks that could appear from the use of a catalogue of RPs), the respondents were giving only their opinion as RE experts, so it cannot be assumed that every benefit and drawback that was marked as relevant in the results would actually appear when using RPs.

Face validity. Face validity is the extent to which a measure addresses the desired concept, i.e. whether it measures what it is supposed to measure. In order to ensure face validity, it was discussed with LIST whether the proposed survey questions were a good representation for answering RQ 1.2. The discussion indicated that the initial set of proposed questions seemed to be suited for answering RQ 1.2.

Since by their very nature questionnaires are rigid and biased instruments due to the predefined options that most questions offer, the most common answers were tried to be covered in the questions that had a list of predefined values (single-answer, multiple-answer, or Likert scale questions). To do this, the values were collected from the principal books and publications on RE (e.g. Hull's et al. [1] and Pohl's [13] RE books). In addition, it was always possible to include alternative choices that were not explicitly offered by using open value fields.

Finally, for the analysis of questions using Likert Scales, even though calculating the average is a controversial issue in the case of ordinal scales (as presented by Boone and Boone [152]), it was decided to calculate it: the scales used in the survey were considered not to be completely ordinal but more of the type interval (for which the average may be calculated). This is corroborated by Jamieson [153], where it is explained that when Likert scales (even if they are ordinal) are used in a series of items that when combined measure a particular trait, it is possible to use the average to present the results. In the current study, the particular traits would be, for instance, the types of requirements more likely to be reused or the most common problems in RE that could be mitigated by the existence of an RP catalogue.

2.3.9 Conclusions of the Study

Through Section 2.3, it has been presented an analysis of the results of a survey which goal is knowing the state of the practice on requirements reuse and the possible advantages, success factors and barriers of implementing requirements reuse using RPs. The analysis takes into consideration the 71 valid and complete responses that come from IT professionals with some experience as requirements engineers in industry.

The observations obtained from the analysis of the survey results may be interesting for both RE practitioners and researchers. For RE practitioners, the relevant observations are:

- **[RQ 1.2.1: Reuse Level vs. Reuse Techniques]** Since the results show a relationship between the level of reuse and elaborated reuse techniques, Table 2.28 could guide organizations that want to choose a requirements reuse technique or improve the requirements reuse technique they are currently using.

- **[RQ 1.2.1: Types of Requirements likely to be Reused]** The survey results related to the types of requirements that are more likely to be reused indicate that a requirements reuse adoption endeavour should first consider reliability, maintainability, usability and security requirements. Companies that usually work in product lines or projects within a same domain or context might also consider reusing requirements in functionality chunks that are present in all of their projects.
- **[RQ 1.2.3: Problems Mitigated by Reuse]** The respondents agreed on the fact that RPs could help in improving the uniformity, lack of ambiguity, and lack of consistency of requirements specifications, and also in the completeness of requirements on a certain non-functional or functional aspect. Therefore, this technique could be interesting for companies that want to improve the quality of requirements specifications.
- **[RQ 1.2.3: Critical success factors and barriers]** Based on the main success factors and barriers identified by the participants, requirements reuse based on RPs should not be introduced in an organization without the agreement of the requirements engineers. In addition, the results show that it would be advisable to define a reuse method that is simple to integrate into the existing software engineering processes of the organization together with a tool to support it.

For RE researchers, the relevant observations are:

- **[RQ 1.2.2: Problems on Reuse Proposals]** There should be greater dissemination of requirements reuse techniques and requirements reuse process adoption, and more studies about the experiences on adoption of these techniques. More specifically, the results of these experiences in terms of return-on-investment are really important in order to convince companies of the benefits of introducing a requirements reuse technique.
- **[RQ 1.2.3: Problems mitigated by Reuse]** Research in how to improve requirements quality by reusing requirements must continue. The same is true for research in applying reuse proposals in organizations.
- **[RQ 1.2.3: Critical success factors and barriers for Reuse]** The proposals of well-defined reuse methods and tools should be formulated in order to achieve methods and tools that are suitable for, and accepted by, organizations.

2.4 General Conclusion

As the results of the SMAP presented in Section 2.2 show, there are multiple proposals for reusing requirements through patterns during the elicitation and specification of requirements, which differ in different aspects. However, most of these proposals are fuzzy, in the sense that they define precisely the pattern approach but they do not take into account other assets that are needed to integrate the proposal in real environments. Specifically, the critical points that are not covered by more than half of the proposals of this SMAP are: establishing relationships among the artefacts to reuse; arranging and classifying them to ease their access; incorporating a metamodel of the artefacts to reuse and their arrangement; proposing methods to construct and update the artefacts; defining a catalogue of artefacts general enough and finished so it can be used as a base example; implementing tools to facilitate the incorporation of the proposal in real environments; and testing the proposal in real environments, proving its validity and the economic benefits that the use of the proposal may entail.

The SMAP also shows that there is not any widespread proposal used in the organizations, nor any complete proposal with a framework that covers all the necessary elements to encourage organizations to adopt requirements reuse.

The last statement is reinforced by the results of the survey-based empirical study presented in Section 2.3. The empirical study concludes that requirements reuse is not an established practice in IT organizations and that most of the people reusing requirements is doing it in ad-hoc manner. Specifically, a very small percentage of the participants is using a patterns-based approach for requirements reuse, although the study reveals that using elaborated requirements reuse techniques drives to higher levels of reuse. The empirical study also identifies that the two most critical aspects for successfully adopting requirement patterns-based approaches are the existence of a reuse method specific for the approach as well as having a tool that supports it. Finally, the empirical study also shows the general belief that patterns-based approaches for requirements reuse could mitigate problems related to the quality of requirements specifications. In addition, participants using some kind of requirement pattern think that this kind of patterns could help to spend less time in the elicitation and specification of requirements.

Altogether, these facts reinforce the motivation behind this thesis, giving justification to the development of the approach presented in it (represented by P21 in the SMAP), which corresponds to a complete framework for requirements reuse through patterns during the elicitation and specification stages, guiding organizations during the reuse process in order to improve their RE processes.

CHAPTER 3 Overview of the PABRE Framework

3.1 What is PABRE

The PABRE framework proposes the use of SRPs to capture and use requirements knowledge during the elicitation and specification of requirements in the context of IT projects.

The PABRE framework distinguishes two types of roles in regard of how they work with SRPs. On the one hand, as Young states [154], requirements analysts are requirements engineers that elicit, analyse, validate, specify, verify, and manage the real needs of IT projects; therefore, they use SRP catalogues to elicit and specify requirements. On the other hand, RE experts are not only specialists on RE, but also on SRPs; therefore, they are the ones in charge of creating, maintaining and updating SRP catalogues.

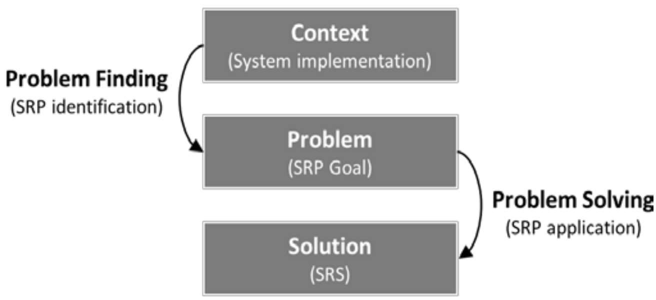


Figure 3.1 – SRPs in the PABRE framework

SRPs follow the typical context-problem-solution structure of patterns (Figure 3.1). During the requirements elicitation of an IT project, a requirements analyst and a customer agree on the requirements of the system under development. In the PABRE framework this is done with the help of the SRP attributes and the SRP classification schemas. In this regard, the main attribute of an SRP used during this identification is the goal, which summarizes the problem that the customer wants to solve. Later on, the problem is solved by adding requirements to the SRS of the system under development.

These requirements are stated from the application of the identified SRPs, as SRPs contain natural language requirement templates that relate to the solution of the goal.

The framework embraces (Figure 3.2): 1) A metamodel that describes the structure and semantics of SRPs, the relationships among them, and their organization into an SRP catalogue (Chapter 4); 2) Methods for guiding the use of the SRP catalogue during requirements elicitation and specification (Chapter 5), as well as other ones for constructing and updating it (Chapter 6); 3) An SRP catalogue composed by 28 NF-SRPs, 38 NT-SRPs and 44 F-SRPs (Chapter 7), the functional ones being specific for the Content Management System (CMS) domain; 4) An economic model to perform cost-benefit analysis on the adoption of SRPs based on ROI (Chapter 8); and 5) The PABRE system as technological support (Chapter 9). The framework has a web resource dedicated to it accessible at http://www.upc.edu/gessi/PABRE/.

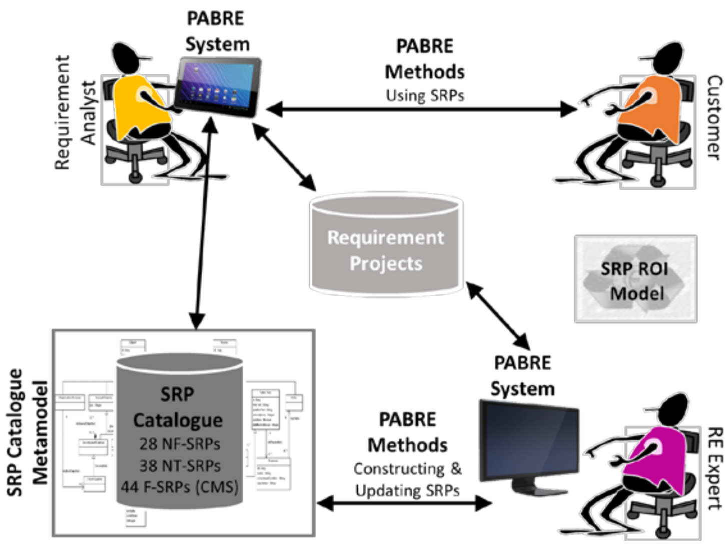


Figure 3.2 – PABRE framework overview

In the following, Section 3.2 presents the concept of SRP in the PABRE framework by means of an example, and Section 3.3 the organization and classification of SRPs in a catalogue. Section 3.4 and 3.5 explain the potential benefits and drawbacks of the framework, respectively.

3.2 SRP in PABRE

An SRP, as used in PABRE, is a pattern that groups a set of requirements (converted into templates) that together pursue a given goal in the system under development. When applied in a specific project, the templates of an SRP produce software requirements that solve the goal specified in that pattern. Aside from these templates and the goal, SRPs have other attributes that help during the application of the pattern (like keywords and information of the parameters on the templates). There is also some metadata that is worth keeping for management purposes (i.e. author, date, sources, and version). In this section, the concept of SRP is presented by showing an example that facilitates the understanding of the structure of SRPs in the PABRE framework. In Chapter 4 this structure is justified and rigorously stated.

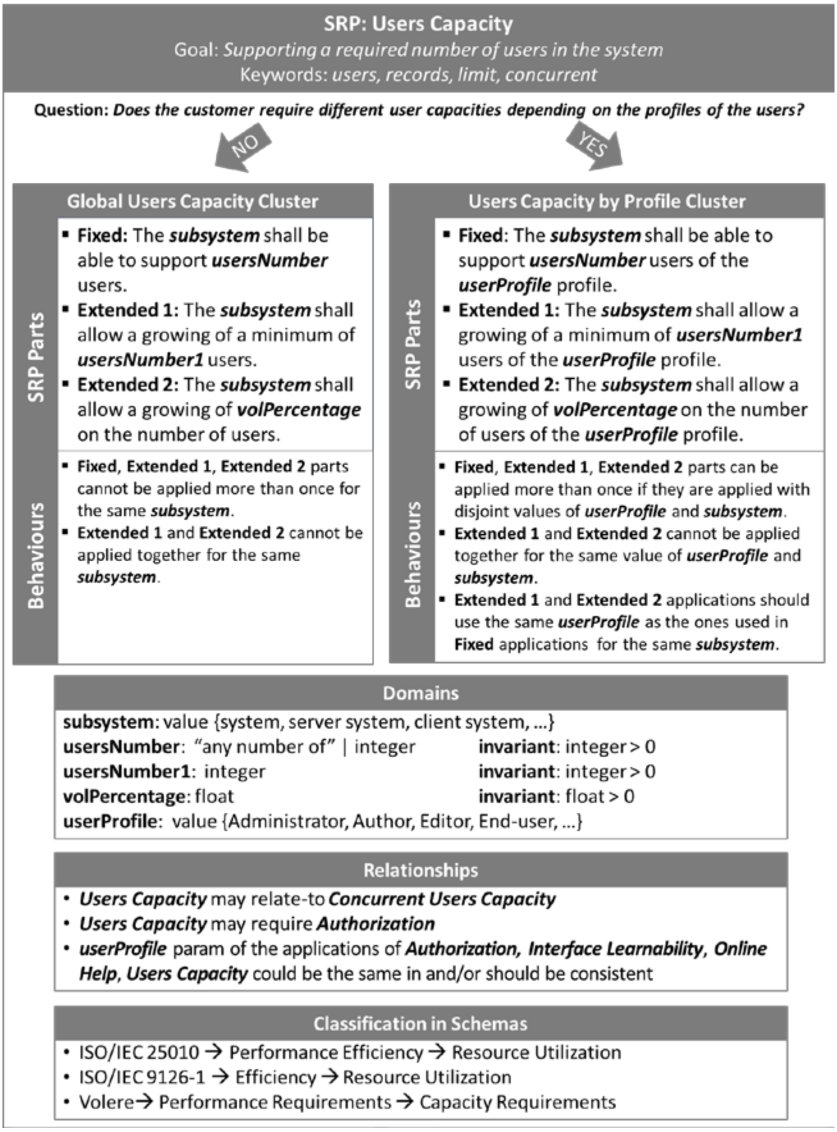


Figure 3.3 – Users Capacity SRP

The SRP used as example is the *Users Capacity* pattern (see Figure 3.3), which is part of the SRP catalogue presented in Chapter 7. In the figure it is possible to see the main attributes of the SRP that are relevant for its identification and application in an IT project. The figure also shows the relationships among the *Users Capacity* SRP with other patterns (*Relationships* part) and how this SRP is classified (*Classification in Schemas* part) (for more information about classifications see Section 3.3). Other metadata of the SRP, which are not relevant for the SRP identification and application, have not been included. These metadata are: the pattern's author, the date of last update of the pattern, the sources that induced to think about the convenience to include the pattern in the catalogue, the current version of the pattern, and the textual descriptions of each one of the components that are part of the pattern.

SRPs are used during the elicitation and specification of requirements. Imagine that a customer needs a system with a minimum number of users. Then the *Users Capacity* SRP is selected, as its *Goal* indicates that the application of the pattern will produce requirements that achieve the goal of S*upporting a required number of users in the system*. Goals, therefore, correspond to the problems to be solved by applying the SRP.

The SRP goal can be achieved by applying one of the *Clusters*. An SRP consists of several clusters, each one representing a different solution for accomplishing the goal. In the example, the goal can be attained by defining the users capacity per user profile (*Users Capacity by Profile* cluster), or by defining the global capacity of users, i.e. without taking into account the different types of users in the system (*Global Users Capacity* cluster). To facilitate the selection of one cluster or another, SRPs include a *Question* whose answer points out the cluster that best applies.

The clusters are organized into a *Fixed Part* and zero or more *Extended Parts*, each of them being a sentence template. The fixed part characterizes the cluster and it is always included in the SRS if the cluster is chosen; due to this nature, the fixed part is usually quite generic and vague in how to achieve the goal. The extended parts are used if more precise requirements are required, so they may be applied or not. The parts are closely related to the *Domains* of the parameters, as the templates are natural language requirements that contain parameters to be instantiated when applied to projects, and the domains of the parameters establish the set of possible values that the parameters can take and eventually their correctness condition (*Invariant*). In the example, the fixed part of the first cluster is *The %subsystem% shall be able to support %usersNumber% users* (*subsystem* will be replaced by the system to which the requirement apply, and *usersNumber* will be substituted in applying the SRP by "any number of" or by an integer greater than 0). The extended parts of the same cluster allow specifying the growth in number of users of the system. The first part states the growth by amount: *The %subsystem% shall allow a growth of a minimum of %usersNumber1% users* (*subsystem* as above, and *usersNumber1* will be substituted in applying the part by an integer greater than 0). The second part states the growth of users by percentage: *The %subsystem% shall allow a growth of %volPercentage% on the number of users* (*subsystem* as above, and *volPercentage* will be substituted in applying the part by a float greater than 0).

Usually, fixed and extended parts must conform to some *Behaviours* for declaring multiplicities or dependencies among parts. In the *Users Capacity* SRP, aside from restrictions on the possible number of appearances of each part in a specific SRS, there exist restrictions on the parameters' values in each application. For instance, in the second cluster of the example, the fixed part can be applied more than once in an SRS as long as the values assigned to the parameter *userProfile* and *subsystems* are different. This allows to state restrictions on the users capacity of a specific subsystem for different types of profiles, such as *Administrator* or *End-User*.

There exist relationships among SRPs in the same way as they exist among requirements. The *Users Capacity* SRP is involved in two relationships with other SRPs. The first one with the *Concurrent Users Capacity* SRP: there is a clear relationship between the number of users to support and the number of concurrent users to support. This means that, when applying the *Users Capacity* SRP, the requirements analyst has to be aware that the two numbers are not contradictory, for instance requiring a number of concurrent users higher than the number of supported users. The second relationship is with the *Authorization* SRP, since the *Authorization* SRP allows defining the user roles supported by the system under development, and the parts of the *Users Capacity per Profile* cluster have as parameter the user roles. This means that the requirements analyst has to be aware that, if *Users Capacity per Profile* cluster of the *Users Capacity* SRP is used, the *Authorization* SRP has been or will be used, as it does not make sense to define the users capacity per user role if roles are not supported by the system under development.

Finally, there are also relationships that allow maintaining the consistency among the requirements in an SRS. In the case of the example, this relationship is based on the fact that the *Users Capacity*, *Authorization*, *Interface Learnability* and *Online Help* include in their definition the parameter *userProfile*. The relationship is an advice to the requirements analyst of maintaining the consistency among the values of the parameters in case the four SRP are applied in the same project.

3.3 SRP Catalogue in PABRE

The existence of SRPs by themselves does not ensure an efficient implementation of requirements reuse. It is necessary to set up an infrastructure able to support the requirements analyst to organize and apply them. The PABRE framework copes with this aspect through a catalogue of SRPs.

An SRP catalogue consists on a repository that stores a collection of SRPs and the possible classification to organize such collection. Although the attributes of the SRPs and relationships among them allow browsing the collection of SRPs in multiple ways, it is advisable to have classifications of SRPs over some criteria. These possible classification are called *Classification Schemas*. In addition, the classification of SRPs can be taken as the organization of requirements in an SRS. It is important to observe that different contexts (organizations, projects, standards, etc.) may, and usually do, define or require different classifications for organizing the requirements in an SRS, as corroborated by Pohl and Rupp [6]. Thus, trying to impose a particular classification schema is a stiff approach. For this reason, PABRE decouples SRPs from classification schemas (see Figure 3.4): the latter just impose different structuring schemas on top of the former.

Inside a classification schema, SRPs are bound to *Basic Classifiers* (the lower level of classifiers). Basic classifiers can contain more than one SRP classified on them. Therefore, they allow joining the SRPs that may be applied as a group, and that address the same functionality or describe the same regulation required in the new system. These basic classifiers are organized into *Compound Classifiers*, which impose the usual hierarchical structure of any classification schema. In order to not impose unnecessary constraints that could lead to rigidness, an SRP could be bound to more than one basic classifier in a single classification schema, and a classification schema may not cover all existing SRPs (i.e. some SRPs may not be classified).

As an example, the SRP in Figure 3.3, in its *Classification in Schemas* section, establishes the classification of the pattern in three different classification schemas. One that is based on the

ISO/IEC 25010 product quality model standard [132], one based on its previous version (i.e. ISO/IEC 9126-1) [137], and one based on the Volere approach [155]. Having the catalogue classified regarding the two standards, makes it usable for organizations that use any of them.

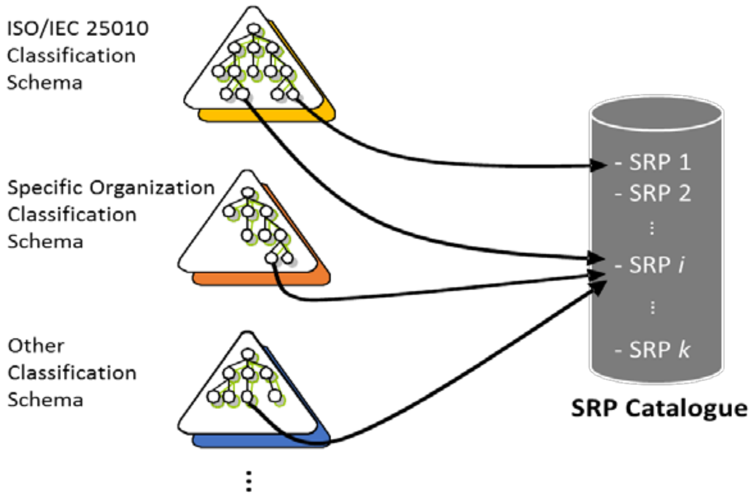


Figure 3.4 – SRP classification schemas

3.4 Possible Benefits of PABRE

The potential benefits of the PABRE framework are the reduction of time spent to perform the elicitation of requirements and the improvement of the quality of the SRS obtained. In this section these benefits are described in an intuitive way. Chapter 11 of this thesis provides evidence based on an empirical study.

Faster requirements elicitation and specification process. With the SRP-based requirements elicitation and specification processes proposed in the PABRE framework, the aim is to downsize the time needed during these stages.

Such reduction of time comes from the fact that SRPs offer "ready-to-use" requirements and that the SRP catalogue can cover the most recurrent requirements (e.g. requirements for different projects addressing the same domain, requirements addressing functionalities shared by different systems, and requirements addressing certain regulations). In addition, the SRP catalogue and the usage methods have been designed to choose requirements in a faster way, since the most frequent output (requirements that are the application of SRPs) has the shortest decisional path. Because of this reduction in time, more time will be available for the definition of creative requirements (the ones that may change the typical behaviour of a system and that provide them an added value).

Improved quality of SRSs. The ISO/IEC/IEEE 29148 standard [156] (predecessor of the IEEE 830 standard for recommended practices in SRSs [138]) describes recommended practices for the processes and systems related to the engineering of requirements for systems and software systems as well as services throughout the lifecycle. Among the recommendations, it defines the characteristics that good requirements should have. These characteristics are, when thinking about requirements as an individual asset: *Necessity*, *Implementation Free*, *Unambiguity*, *Consistency*, *Completeness, Singularity*, *Feasibility*, *Traceability*, and *Verifiability*. When thinking about requirements as a group, the characteristics considered in the standard are: *Completeness*, *Consistency*, *Affordability*, and *Bound*. Below, it is justified how the use of the PABRE framework may drive to good requirements taking into account the characteristics of this standard (Table 3.1). The analysis takes into account only those requirements coming from SRPs, and not the whole produced SRSs, since it is not possible to guarantee these characteristics for the new requirements that appear in SRSs (i.e. those requirements that do not come from applying an SRP).

The analysis of the characteristics for individual requirements is presented below.

- **Necessity.** The use of SRPs cannot guarantee that the elicited requirements are essential for the new system. However, the characteristic also states that requirements should be applicable (e.g. using technologies that currently exist) and not obsolete. This part is addressed in the SRP catalogue, since it will be constantly evolving (including new technologies and removing obsolete elements when necessary).
- **Implementation Free.** Because of the participation of RE experts while defining SRPs and the quality check that is done at the end, it could be guaranteed that the requirements coming from them state what is required for the system, but not how the requirement is met.
- **Unambiguity.** The idea is that the SRP catalogue will be unambiguous, since every SRP will have this quality no matter the SRSs they have been extracted from (i.e. even if the SRP is extracted from ambiguous SRSs, the SRP will not be ambiguous). This will be achieved by having RE experts that carefully review the

texts in the SRPs, and having a glossary of terms usually appearing in the SRP catalogue (which also contains a list of synonyms for each term). This processing could eventually guarantee that the requirements extracted from applying SRPs will very rarely have any ambiguity.

Table 3.1 – ISO/IEC/IEEE 29148 good characteristics for requirements addressed by PABRE

ISO/IEC/IEEE 29148 Characteristics for Good Requirements			Addressed
Individual Requirements	**Necessity**	The requirement defines an essential capability, characteristic, constraint, and/or quality factor. The requirement is currently applicable and has not been made obsolete by the passage of time.	**Partially**
	Implementation Free	The requirement, while addressing what is necessary and sufficient in the system, avoids placing unnecessary constraints on the architectural design. The objective is to be implementation-independent. The requirement states what is required, not how the requirement should be met.	**Yes**
	Unambiguity	The requirement is stated in such a way so that it can be interpreted in only one way. The requirement is stated simply and is easy to understand.	**Yes**
	Consistency	The requirement is free of conflicts with other requirements.	**Yes**
	Completeness	The stated requirement needs no further amplification because it is measurable and sufficiently describes the capability and characteristics to meet the stakeholder's need.	**Partially**
	Singularity	The requirement statement includes only one requirement with no use of conjunctions.	**Yes**
	Feasibility	The requirement is technically achievable, does not require major technology advances, and fits within system constraints (e.g. cost, schedule, technical, legal, regulatory) with acceptable risk.	**Partially**
	Traceability	The requirement is upwards traceable to specific documented stakeholder statement(s) of need, higher tier requirement, or other source (e.g. a trade or design study). The requirement is also downwards traceable to the specific requirements in the lower tier requirements specification or other system definition artefacts.	**Partially**
	Verifiability	The requirement has the means to prove that the system satisfies the specified requirement. Verifiability is enhanced when the requirement is measurable.	**Partially**
Set of Requirements	**Completeness**	The set of requirements needs no further amplification because it contains everything pertinent to the definition of the system or system element being specified.	**Partially**
	Consistency	The set of requirements does not have individual requirements which are contradictory. Requirements are not duplicated. The same term is used for the same item in all requirements.	**Yes**
	Affordability	The complete set of requirements can be satisfied by a solution that is feasible within lifecycle constraints (e.g. cost, schedule, technical, legal, regulatory).	**No**
	Bound	The set of requirements maintains the identified scope for the intended solution without increasing beyond what is needed to satisfy user needs.	**No**

- **Consistency.** The SRP catalogue will be processed by RE experts to guarantee that the wording in a requirement is consistent, by using a glossary that includes a list of synonyms of terms usually appearing in the SRP catalogue. In addition, during this process, any dependency that may exist among SRPs will be explicitly stated in the SRPs. If these relationships are taken into account during the elicitation and specification process, it is possible to guarantee that the subset of requirements of the SRSs that have been extracted from the catalogue are consistent.
- **Completeness.** The individual requirements of SRPs are complete in general, since again they are the result of the study of multiple SRSs after RE experts review and rewrite them. However, SRP fixed parts are usually abstract, so it may be difficult to measure them, hence these parts are not complete.
- **Singularity.** The organization of SRPs into parts, and the quality check done at the end of the construction phase of the catalogue, takes cares of assuring that requirements elicited using SRPs contain one, and only one, requirement.
- **Feasibility.** The update process of the SRP catalogue guarantees that the requirements are technically achievable and it does not require major technology advances, since only technologies currently available will be part of the SRP catalogue. However, the use of the catalogue cannot assure that a requirement fits within system constraints and acceptable risk.
- **Traceability.** On the one hand, the upwards traceability of requirements that were extracted from the SRP catalogue is partially guaranteed because the catalogue includes the sources from where the SRPs were derived. However, the other references to the origin sources do not depend on the use of the catalogue. On the other hand, the downwards traceability to requirements in lower tiers does not depends on the use of the SRP catalogue neither.
- **Verifiability.** The way in which SRSs are wrote will drive to verifiable requirements that do not contain any non-definable or non-evaluable terms. Specifically, when an SRP is applied, the parameters of the SRP must take some concrete value that will ensure the verifiability of the corresponding requirements. However, some fixed parts of the SRP catalogue are abstract and therefore difficult to be measured.

As for sets of requirements, the analysis of the characteristics pointed out by the standard is:

- **Completeness.** Providing a complete SRP catalogue may contribute to obtain complete SRSs. This could be possible after some time of using the catalogue, and once a stable version of it is achieved. However, it is worth taking into account that a strictly complete version of the SRP catalogue will never arrive, since there may always exist the need of requirements that are very specific of projects, and it would not have sense to have SRPs for them.
- **Consistency.** As well as the consistency in individual requirements, the inclusion of not contradictory requirements is guaranteed by the relationships established between SRPs, provided they are used while eliciting and specifying the requirements. Requirements cannot be duplicated since the methods to use SRPs have rules to control that aspect, and the consistency of terms is assured because a glossary of terms, including a list of synonyms for each term, is used while constructing the SRP catalogue, so SRPs are always using the same term when referring to the same concept.
- **Affordability.** The use of SRPs cannot warrant that the elicited requirements are affordable, since it is not possible to control if the requirements elicited applying SRPs are obtainable within the lifecycle constraints.
- **Bound.** Although usually SRPs will be used in meetings with customers and they will give their opinion on the requirements that are being elicited, the use of SRPs per se does not assure that the elicited requirements are within the scope of the new system.

3.5 Possible Drawbacks of PABRE

There are several factors that may compromise the use of the PABRE framework, some of them highlighted in the empirical study presented in Chapter 11. In the following, these factors are described in an intuitive way.

The integration of the requirements reuse method. Some changes are necessary in the RE processes of organizations in order to integrate SRPs (e.g. a new tool or plugging, or a maintenance process over the SRP catalogue). In addition, the willingness of requirements engineers to apply the reuse and adapt their processes to SRPs can be hazard, since they are used to work in a specific way and that is somehow difficult to change.

The cost of maintenance of the SRP catalogue. On the one hand, in order to have an SRP catalogue that is valid in the long term, it is necessary to have someone (either a person, a group of them, or a department) that is responsible of the maintenance of the catalogue. This responsibility implies taking care of the catalogue update, and maintaining its quality or even improving it, so the catalogue does not become obsolete or degrades over time. On the other hand, the return of investment in the maintenance of the SRP catalogue has to be proved to IT CIOs, since it could help to convince them to invest in the use of SRPs.

The "heaviness" of the reuse process. The main process for eliciting and specifying requirements using SRPs may be "heavy" for inexperienced requirements analysts that discover the SRP catalogue and that are more used to collect requirements in a less driven manner. It is then necessary to plan an initial training on the concept of SRP and on the navigation throughout the catalogue. For this last necessity, goal matching or faceted descriptions using keywords could be used.

The "inefficiency" of the reuse process. Requirements analysts can find "inefficient" the fact of processing the entire catalogue during one interview rather than identifying requirements in an exploratory way. To tackle this issue, different options for using the SRP catalogue, ignoring the main method to use SRPs, are proposed. As a summary, these options are: 1) Pre-selecting SRPs before interviewing customers; 2) Using the SRP catalogue with a checklist character; and 3) Using natural language matching techniques to propose SRPs while requirements analysts type requirements. More information about these possibilities can be found in Subsection 5.2.2.

The context of use of an SRP catalogue. This factor is a challenge in case the SRPs are constructed and maintained by a different team than the one is using it, or by a different organization. The success would depend, for instance, on the agreement in terminology among the teams, making in this specific case the importance of the glossary of terms even bigger.

SRP impact in innovation. SRPs are based on reusing consolidated knowledge, therefore requirements analysts may rely too much on them, hindering the elicitation of innovative requirements. A right balance is therefore needed. As a consequence, it is necessary to explain to requirements analysts the kind of requirements they can expect to find in the SRP catalogue.

SRP impact on requirements. The way of using SRPs will be ultimately decided by the requirements analysts. Some approaches to use SRPs (such as the pre-selection of SRPs presented in Subsection 5.2.2) may end with customers accepting requirements that are not really of interest to them just because the requirement analyst proposed the requirements. Therefore, SRSs would actually contain more requirements than necessary, with the impact that this factor entails. To mitigate this risk, it is necessary that not only the customer but the requirements analyst makes sure that a requirement is actually needed in an IT project.

CHAPTER 4 Metamodel of an SRP Catalogue

4.1 Introduction

The structure of SRPs and consequently their metamodel was obtained using three types of knowledge: SRS documents, knowledge of experts and literature about requirements and requirement patterns.

The applicant grounded her theory [157] mainly on SRS documents. These SRSs came from call-for-tender projects undertaken by LIST, one of the organizations that is collaborating with GESSI. These SRSs are focused on the domain of CMSs. The observations that drove to the SRP structure were done on the Non-Functional Requirements (NFRs) of 6 SRS documents, and afterwards validated regarding Non-Technical Requirements (NTRs) and Functional Requirements (FRs). This validation can be found in Chapter 10 of this thesis.

After the SRSs observations, three types of experts were involved in validating the consequences related to the SRP structure derived. These experts were the GESSI academic-experts with a good background on several aspects of RE and metamodelling; LIST experts with high practical experience in the statement of requirements expressed in natural language; and a Luxemburg local network of IT consultants with a RE profile experienced in off-the-shelf based selection processes.

Also, three types of publications were consulted in order to support the statements and observations on the structure of requirements and consequently on the structure of SRPs. The used literature was: classical software requirement textbooks and papers; SRP-focused literature; and specific literature about particular types of requirements.

This chapter presents the statements, the observations and the consequences, explaining the experts' opinions and indicating literature that supports them. The resulting metamodel is modelled using the rigor of UML language. For a matter of ease in the explanation, the metamodel has been divided into three parts: the core structure of SRPs (presented in Section 4.2), the relationships among SRPs (presented in Section 4.3), and the classification of SRPs (presented in Section 4.4). The metamodel as a whole as well as the glossary of all the concepts appearing in the metamodel can be found in Appendix III.

4.2 Core Structure of SRPs

The analysis of SRS documents is articulated through a series of relevant statements supported by observations that have consequences on the metamodel (Subsection 4.2.1). After that, the final version of the metamodel is presented (Subsection 4.2.2). As starting point, it was considered that the metamodel contains a class *Software Requirement Pattern* (*SRP* for short).

4.2.1 Analysis of SRS Documents

Statement 1

> *A requirement refers to a concept. In different SRS documents, some of these concepts appear repeatedly.*

Observations. The analysis of SRS documents shows that different requirements in different SRSs establish constraints about the same concepts, as for example constraints on *Authorization*, *Concurrent Users Capacity*, *Interface Language*, etc. Specifically, the requirements in the 6 SRS documents, which sum up 533 NFRs, refer to 46 different concepts. Figure 4.1 and Figure 4.2 show the concepts that were identified and the number of SRSs where requirements referring to each of them were found. For instance, requirements referring to the concept *Platform* appear in all the SRS documents. In some cases, there was a doubt about to which concept the requirement was referring, having more than one concept as option. For instance, two concepts were considered as an option for Req-11 in Table 4.1, the *Update Procedures* of the system and the *Versioning* of the system, but a careful look to the context where the requirement appeared clearly showed that the concept being referred by the requirement was *Update Procedures*.

Initially, it was supported the idea that requirements like those for *Platform* (see eight first rows in Table 4.1), which referred to a concept that appeared in more than one SRS, could be considered the potentially reusable requirements and logical candidates to appear in future projects. This condition applied to 31 out of the 46 concepts. However, experts' opinion did not completely agree with this assumption. Specifically, they considered that:

Table 4.1 – Excerpt of requirements found in the analysed SRS documents (I)

Req. Id	Concept	Requirement	Project Entity	SRS Id
Req-1	Platform	The **system** should be based on a **standard web server**.	Whole system, Web server	SRS-6
Req-2	Platform	The **system** should be based on (or be integrated with) a **standard mail server**.	Whole system, Mail server	SRS-6
Req-3	Platform	The **system** should be based on a **management system standard database**.	Whole system, Database	SRS-6
Req-4	Platform	The **client modules** must be portable across **Windows2000 or higher, GNU / Linux (Suse and Debian) and Mac OS X operating systems.**	Client subsystem, Operating systems	SRS-3
Req-5	Platform	The **client web browsers** should be **Internet Explorer 5, Firefox 1.5 +, Konqueror and Safari**.	Client subsystem, Web browser	SRS-3
Req-6	Platform	The **system** should use **Oracle**.	Whole system, Database	SRS-3
Req-7	Platform	The **system** must run on an operating system under **Windows (NT/2000/XP)**.	Whole system, Operating system	SRS-2
Req-8	Platform	The **proposed system** should work with a **relational database manager**.	Whole system, Database	SRS-2
Req-9	Update Procedures	The **system** shall have automatic update processes.	Whole system	SRS-3
Req-10	Update Procedures	The update process should start **checking if a later ve-rsion exists and offering it to the user if it is the case**.	Whole system	SRS-3
Req-11	Update Procedures	If updated, the changes made by the new version are shown to the user.	Whole system	SRS-3
Req-12	Ease of Use	The **solution** should permit **navigation from a field to other in the same screen only with the keyboard**.	Whole system	SRS-1
Req-13	Ease of Use	**Keyboard shortcuts** are desired. Their parameterization by the user will be a plus.	Whole system	SRS-5

- On the one hand, some requirements aligned to concepts found in more than one SRS were considered non-reusable (see the 5 leftmost bars in Figure 4.2). Remarkably, this happened to 3 concepts that appear in 5 of the 6 SRSs. Experts considered that their aligned requirements were too specific to become an SRP, or that their low quality would made the conversion impossible, or even that they were not proper requirements. For instance, the two requirements referring to the concept *Ease of Use* shown in Table 4.1 were considered too specific for an SRS, since they deal with details on the keypad use or the fields' forms tabbing.
- On the other hand, some requirements referring to concepts found in just one SRS were considered reusable (see the 2 rightmost bars in Figure 4.1). For instance, the requirements aligned to the concept *Update Procedure* (see Table 4.1) were considered interesting for the SRP construction, since they express the need of stating how the system updates shall be done when new versions of it appear. Experts considered that the reason they did not appear more often in past

projects was probably an oversight to avoid in the future. The rest of requirements aligned to a concept appearing in just one SRS (13 concepts) were considered as non-reusable for the same reasons enumerated above: being too specific or low quality requirements.

The idea of requirements referring to concepts also appears in other RE works, as in Ali and Kasirun's work [158], that talks about requirements being structured into concerns, or in Alencar and Fernandes' proposal [159], where the first step for identifying crosscutting concerns is extracting the relevant topic, which they call code, for each requirement.

Consequence. Concepts are explicitly represented in the metamodel (*Concept* metaclass). A one-to-many binary association among *SRP* and *Concept*, *refers-to*, keeps track of the concept each SRP refers to. ■

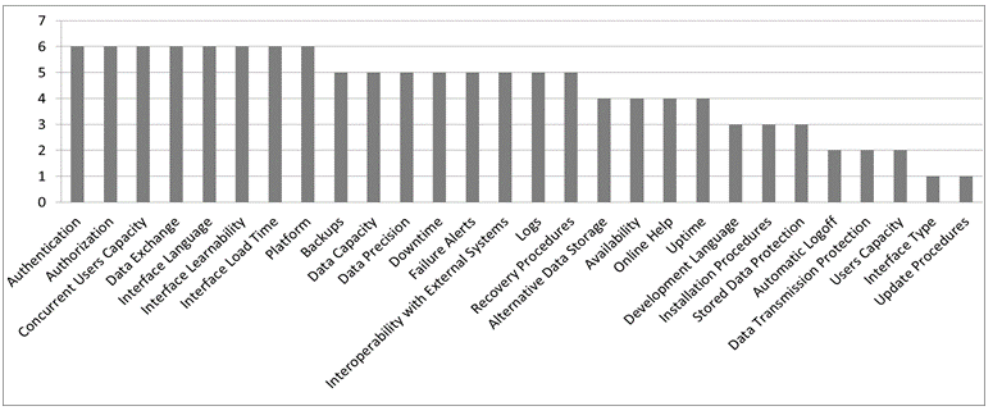


Figure 4.1 – Number of SRS documents per concept related to reusable requirements (1)

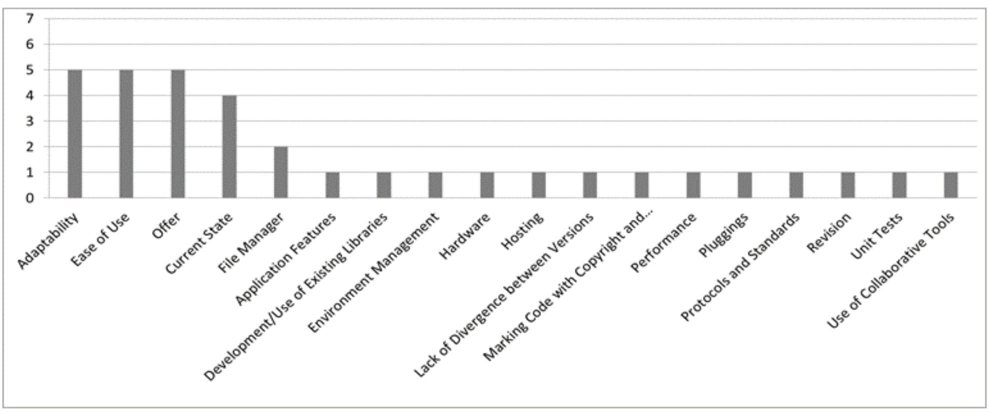


Figure 4.2 – Number of SRS documents per concept related to non-reusable requirements (2)

NOTE: Hereinafter, the analysis of the SRS documents is focused only on the requirements related to reusable concepts, i.e. 417 NFRs.

Statement 2

A requirement restricts one or more entities related to the project.

Observations. In the analysed SRS documents, most of their requirements are restricting the system or subsystem under specification. This can be seen in Table 4.1. In all the requirements this entity is the *Whole System*, except in Req-4 and Req-5 that it is the *Client Subsystem*. Sometimes this entity is implicit, as it happens in Req-10 and Req-11. Other specific entities can be restricted, as in the *Platform* aligned requirements, where conditions are established on entities of the type *Platform Technology*: *Database*, *Web Server*, *Operating System*, etc. In fact, a requirement can restrict more than one entity of different entity types, as in the case of the *Platform* requirements appearing in Table 4.1, which restrict two entity types, *Subsystem* (considering that the *Whole System* is a particular instance of this type) and *Platform Technology*.

In the 6 SRS documents, 100% of the requirements always restrict implicitly or explicitly some entity. As can be seen in Figure 4.3, 7 types of entities were found, and for each of them it shows in how many requirements they are relevant. All the requirements restrict the *Subsystem* entity, and 98 of them restrict other entities.

In the literature, the IEEE 610-12 [160] explicitly states that “Requirements are conditions or capabilities that must be met or possessed by a system or system component”, corresponding to the *subsystem* entity type identified in the analysed SRS documents. The idea of requirements constraining entities also appears in Pohl and Rupp [6], where they explain that aspects that can be altered during system development are, for instance, business processes, technical processes, roles, and components of the system.

Consequence. Entity types are explicitly represented in the metamodel (*Entity Type* metaclass). A many-to-many binary association among *SRP* and *Entity Type*, *constrains*, keeps track of the entity types that are constrained by every SRP. ■

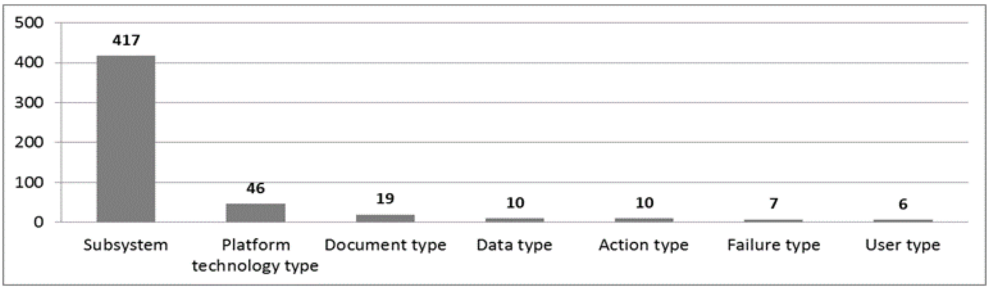


Figure 4.3 – Number of requirements per type of entity

Statement 3

> *Several requirements in a single SRS document refer to the same concept and establish constraints on the same entities. This group of similar requirements is called a requirements cluster.*

Observations. Requirements do not appear as isolated units in a typical SRS, but they present some relationships. In particular, some of them are tightly related and cannot be conceived without each other, since they refer to the same concept and restrict the same entities, forming these requirements clusters. In Table 4.1 the clusters in an SRS document have been signalled with different background colours. For example, in the case of the *Update Procedure* concept and the *Whole System* entity, there is a cluster in the SRS-3 document composed of three requirements. The first requirement (Req-9) requires the existence of an automatic update procedure for the whole system under development; the second and third require, respectively, how to proceed with the updates (Req-10) and the guidance of the user in the changes introduced in the new system versions (Req-11). In the case of the *Platform* concept and the *Subsystem* and *Platform Technology* entity types, different clusters arise. Specifically, in the SRS-6 document there appear three clusters, all of them corresponding to restrictions on the *Whole System* and on a *Web Server*, a *Mail Server* and a *Database* respectively.

In the SRS documents, there are 192 requirements clusters composed of 1 to 9 requirements (Figure 4.4). 54% of them are composed by more than one requirement (see Figure 4.5). In the examples above, the clusters aligned to the concept *Platform* are composed by just one requirement, and the ones aligned to *Update Procedure* by three.

The concept of requirements cluster as a set of requirements that together specify a constraint over the same concept has been widely used in the software engineering field (e.g. the works of Chen et al. [161], Duan et al. [162], and Frakes et al. [163]). However, no literature has been found that supports the fact that these clusters are establishing the constraints on the same entities, which represents a novelty of the proposed SRPs.

Consequence. *SRP* is not simply an abstraction of one requirement, but of a cluster of requirements aligned to one *Concept* and one or more *Entity Types*. Clusters are introduced in the metamodel by means of the class *SRP Cluster*. This metaclass will generate a cluster of requirements when applied in a particular project. Therefore, it is a composition of simpler items, each of them being an abstraction of some requirement. ■

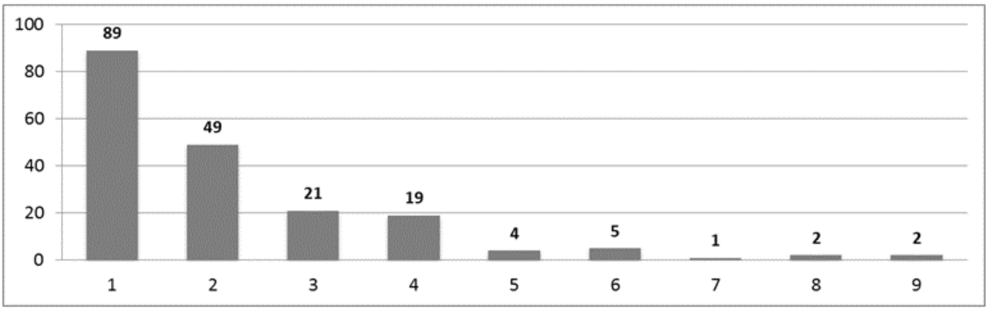


Figure 4.4 – Number of clusters in the SRS documents per number of requirements per cluster

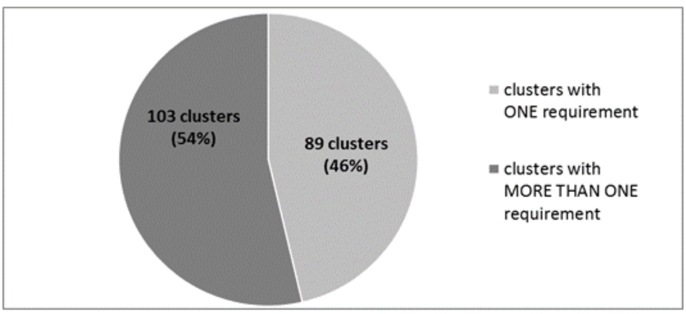


Figure 4.5 – Number of clusters in the SRS documents with more than one requirement

Statement 4

> *One concept and one or more entities of the same entity types may be restricted by different requirements clusters in different SRS documents.*

Observations. The analysed SRS documents show this situation several times. Consider for instance the clusters referring to the *Platform* concept and entities *Whole System* and *Database* (see Table 4.1). Two of them (cluster in SRS-6 composed of Req-3 and cluster in

SRS-2 composed of Req-8) express that the future system has to work on a database that fulfils a specific characteristic (standard, relational). In the other one (cluster in SRS-3 composed of Req-6) a particular database management system (Oracle) is required. In spite of this difference, the goal in both cases is the same, i.e. fixing the database technology; however, they try to achieve this goal in a different way because the context of the project is different.

In 6 out of the 28 reusable concepts different clusters were found for establishing needs or constraints on the same entity types. This corresponds to 21% of the concepts (see Figure 4.6). Table 4.1 shows that the *Platform* concept belongs to this 21%, whilst *Update Procedure* is part of the remaining 79% since its requirements conform just one cluster in one SRS.

It is also important to notice that different requirements clusters referring to the same concept could appear in the same SRS provided that these clusters constrain different entities of the same entity type. For instance, SRS-6 contains three clusters related to the *Platform* concept (see Table 4.1), each of them constraining the *Whole System* entity, but different *Platform Technologies* entities: Req-1 constrain the *Web Server*, Req-2 constrain the *Mail Server* and Req-3 constrain the *Database*. The appearance of more than one cluster for the same concept but different entities of the same entity type in the same SRS was found in 3 reusable concepts *(Installation Procedures*, *Interface Load Time* and *Platform*).

The literature shows different examples about satisfying a goal with different exclusive solutions. The *i** language proposed by Yu [164] includes the constructor *means-ends* which, among other things, allows decomposing a goal into alternative tasks (i.e. solutions) that satisfy the goal. That means that only a task can be chosen to satisfy a specific goal. This same idea can be found in other works dealing with other goal-oriented modelling techniques (e.g. Suppakkul et al. [10] and Mussbacher et al. [165]). Furthermore, in Withall's proposal [9], patterns can contain several alternative templates, each of them tailored to a particular situation.

Consequence. SRPs must be defined to allow generating all these possible clusters. This is done by establishing the one-to-many association *composed_by* between *SRP* and *SRP Cluster* (the metaclass that generates requirements clusters). A derived association, *has_as_entity_types*, is introduced between *SRP Cluster* and *Entity Type*, being *has_as_entity_types* all the *Entity Types* that the *SRP* of the *SRP Cluster* constraints. When an SRP is applied in a project, different clusters can be selected as long as they are applied in different entities. ■

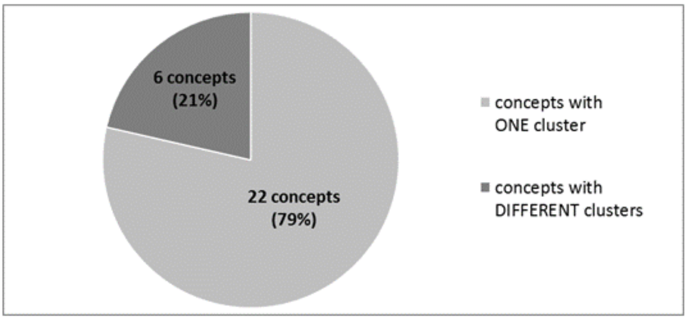


Figure 4.6 – Concepts represented by more than one cluster in the same SRS

Statement 5

> *Each requirements cluster has a requirement that characterizes it, and zero or more requirements that refine the first one.*

Observations. Table 4.1 shows some example of this. For the cluster referring to the concept *Update Procedure* and the entity *Whole system*, the requirement Req-9 characterizes the cluster, whilst requirements Req-10 and Req-11 refine Req-9 and would be meaningless in case Req-9 was not included in the SRS.

This situation occurs always in the analysed SRSs, i.e. for the 100% of their clusters. However, in 26% of the cases, the requirement that characterizes the cluster is implicit. For instance, consider the cluster shown in Table 4.2, referring to the *Authorization* concept in SRS-1. The only requirement appearing in SRS-1 for this cluster is Req-15, which requires the capability of the system to define three concrete profiles at least. SRS-1 does not require explicitly the general capability of defining profiles, which is an implicit requirement (Req-14). If this missing requirement is not fulfilled, the explicit one cannot be satisfied. In other SRS documents, the cluster for *Authorization* includes the requirement explicitly, as in Req-16 for SRS-4 in Table 4.2.

Taking that into account, 100% of the clusters in the analysed SRSs have a requirement that characterizes it. Those clusters that have more than one requirement are those ones that have at least one refining requirement, i.e. the 54% of the clusters (Figure 4.5).

Literature also supports this statement. Hull et al. [1] and Mannion et al. [166] agree that whereas there are requirements that are readable on its own, there are others that make sense only in the context of their "parent" requirement. Smith et al. [167] goes one step further and says that there are "Core" phrases used to express the basic meaning of a system property and perhaps they attach one or more subsidiary phrases used to refine them.

Consequence. An *SRP Cluster* is composed of two different kinds of information, each represented with a new metaclass. First, the *Fixed Part* metaclass, which corresponds to the abstraction of the requirement that characterizes the cluster. In addition, the *Extended Part* metaclass, which corresponds to the abstraction of the requirements that refine the requirement characterizing the cluster. When an SRP *Cluster* is applied in a project for one or more specific *Entities*, which belong to the *Entity Types* to which the *SRP* is associated, one cluster of requirements will be obtained from the application of the parts of the *SRP Cluster*. Since both parts are similar in structure and semantics, they are defined as specialization of the *Requirement Abstraction* abstract metaclass. ■

Table 4.2 – Excerpt of requirements found in the analysed SRS documents (2)

Req. Id	Concept	Requirement	Project Entity	SRS Id
Req-14	*Authorization*	*The **solution** shall allow an administrator to define user profiles that control the authorization mechanisms.*	*Whole system*	***Implicit in SRS-1***
Req-15	Authorization	The **solution** should permit to generate at least three profiles: **administrator, author, and validator**.	Whole system	**SRS-1**
Req-16	Authorization	The **solution** should define profiles for access to the contents, which may be configurable by the administrator.	Whole system	**SRS-4**
Req-17	Authorization	It has to manage at least the following profiles: **Director, Business Manager, Secretariat, other**.	Whole system	**SRS-4**
Req-18	Authorization	The possibility for the administrator to define access permissions for **field and feature** will be a plus.	Whole system	**SRS-4**
Req-19	Authorization	The **solution** should allow to have access to the contents differentiated by the different profiles, which should be configurable by an administrator.	Whole system	**SRS-2**
Req-20	Authorization	It has to manage the following profiles: **Director, Project Manager, Secretariat, other**.	Whole system	**SRS-2**
Req-21	Authorization	Users with profile **Project Manager** should have **full access to all directories of the projects under his responsibility**.	Whole system	**SRS-2**
Req-22	Authorization	Users with profile **Secretariat** should have **read access to all documents**.	Whole system	**SRS-2**
Req-23	Users Capacity	The **system** should work with a **50**% increment in the number of users.	Whole system	**SRS-1**
Req-24	Users Capacity	The **system** should allow a growing of a minimum of **50** users.	Whole system	**SRS-6**

Statement 6

Different requirements corresponding to the same requirements cluster in the same or different SRS documents express the same restriction, but they *also present specific information related to the context of the project to which the SRS corresponds.*

Observations. Requirements Req-15 and Req-17 of Table 4.2 constrain with different sentences the user profiles that the new system would be allowed to create. However, in a second look, the profiles in each requirement are specific of each project (SRS-1, SRS-4). In addition, the entities constrained by the requirements are specific of each project, e.g. the platform technologies and the system part in requirements Req-1 to Req-8 (see Table 4.1). The specific information in these requirements can be seen as a value of a certain domain. As an example, the specific information of requirements Req-15 and Req-17 can be seen as values of the domain of all the possible user profiles a system can have.

Taking this situation into account, requirements stating the same issue, as Req-15 and Req-17, can be generalized to a common schema: "The *subsystem* shall recognize *userProfiles* user profiles.", where *subsystem* is the system or subsystem under specification and *userProfiles* are values of the domain of all the possible user profiles a system can have.

Analysing all the requirements appearing in the SRSs, 89% of the requirements have specific and explicit references to the context of the project. In the requirements of Table 4.1 and Table 4.2, the information dependent on each project is highlighted in bold.

There are other RE works that agree with this statement. Lam et al. [111] introduces the term "template requirement" to talk about requirements expressing the same restriction with different options. The works of Smith et al. [167] and Moros et al. [168] say that different requirement phrases can be grouped to be parameterized, expressing in this way one or more of the options. Withall [9] also states that requirements can contain placeholders for defining variable information.

Consequence. Each *Requirement Abstraction* has a text named *Template*, which may contain references to *Parameters* used when the *SRP* is applied to be adapted to the project context. That *Parameter* is always related to a *Domain* that defines its possible values. ■

Statement 7

The same type of refining requirements may exhibit different behaviours in its appearance in different requirement clusters in the same or different SRS documents.

Observations. Specifically, three behaviours were observed:

- Several appearances of the same type of refining requirement in a cluster. In every appearance, they express the same kind of constraint with a small variation, e.g. in each appearance the requirements restrict different project specific aspects. As example, let us consider the *Authorization* concept. In an excerpt of a cluster related to that concept (see Table 4.2), the requirements Req-21 and Req-22 express the same kind of constraint for different user profiles. As can be seen, this refining requirement could appear in one SRS as many times as user profiles existed. Considering the 54 clusters that have more than one refining requirement in the analysed SRSs, this situation happens in 19 of them.
- The existence of refining requirements in a cluster that do not make sense if another refining requirement is not present in the cluster. As illustration, let us look at the *Authorization* concept requirements appearing in SRS-2 (see Table 4.2): Req-21 and Req-22 express an access constraint for specific user profiles; however, these requirements do not make sense if the specific profiles that the solution has to manage are not stated in any requirement (in the example, these specific profiles are stated in Req-20). This situation appears in 3 of the 54 clusters that have more than one refining requirement in the analysed SRSs.
- The existence of refining requirements in clusters of different SRSs that cannot appear together in a cluster of requirements (exclusion dependency). As example, let us consider the refining requirements for the restriction concept *Users Capacity* (see Req-23 and Req-24 in Table 4.2). They express the required users growing level for a system. It is clear that these refining requirements could not be used together in one SRS document since they define the growing of users in two different ways: the first one with percentage and the second one with an absolute number of users. Of course, this behaviour could not be deduced directly from the SRSs, but it was stated from experts' assessment.

Although no more behaviours were found in the analysed SRSs, expert assessment revealed some potential constraints that could happen in the future, e.g. existence dependency (one refining requirement could be used only if another one is also used).

Literature shows that other works incorporate the idea explained in the statement. Gomaa [169] already talks about mutually exclusive requirements, while Mannion et al.

[166] goes one step further and defines types of discriminants in the requirements that restrict the same concept, among others: 1) mutual exclusion requirements (as Gomaa), but also 2) a list of alternatives requirements where at least one must be selected.

Consequence. The *Extended Parts* of an SRP *Cluster* may present any possible *Behaviour* about their application in a certain project. In some cases behaviours are related to just one part of the pattern and in others may describe a relationship among different parts of the pattern; this is stated in the *Definition* of the *Behaviour*. ■

4.2.2 Metamodel of SRP Structure

Figure 4.7 shows the final version of the metamodel that represents the structure of SRPs, which was mostly derived from the consequences of each statement.

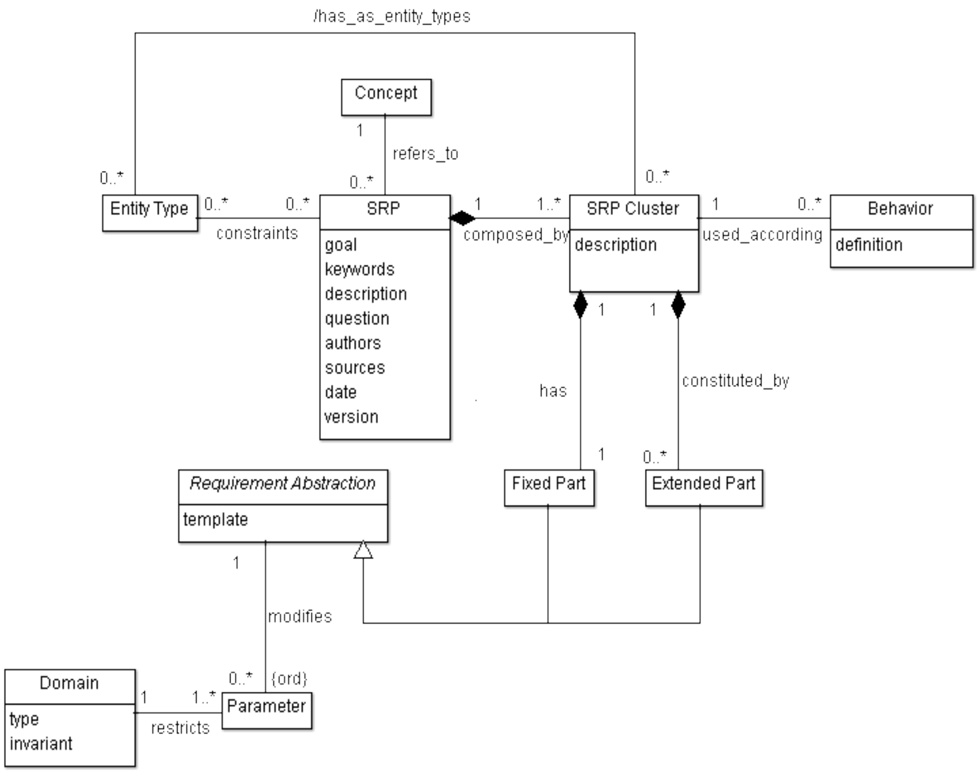


Figure 4.7 – Metamodel: basic structure of an SRP

While validating the metamodel over NTRs and FRs (Chapter 10) it was noticed that some requirements clusters of NTRs did not restrict any entity. Because of that, the multiplicity of the relationships *constraints* and */has_as_entity_types* in the direction *SRP* and *SRP Cluster*, respectively, to *Entity Type* is 0..* instead of 1..* (which is what was deducted from the observations over NFRs in the Statement 3 of the previous subsection). Therefore, *SRP Cluster* will be applied in a project for zero entities, if there is no *Entity Type* associated to it, or for one or more specific *Entities*, if there are *Entity Types* associated to the *SRP Cluster*. It is important to remark the existence of an integrity constraint in this part of metamodel: for all *SRP Cluster* of an *SRP* the *has_as_entity_types* are the same, and they match the *constraints* of their *SRP*.

Finally, the metamodel includes some additional information that was not deduced from the SRSs, but that needs to be added in order to be able to use SRPs, or just to keep track of their evolution. This additional information corresponds to: 1) Some information that helps a requirements analyst to decide whether an SRP is applicable in a project or not, as it is usual in pattern definitions; this data is called the *Goal* and it corresponds to the problem-statement of the pattern. 2) Some significant words in the content of an SRP, called *Keywords*, that are used to index it and facilitate its search; 3) Some statements that correspond to the *Description* of an SRP and its SRP Clusters, i.e. a short representation of what the SRP and SRP Clusters are about, respectively; 4) A *Question* that helps to decide what is the cluster of an SRP that best applies on an IT project (only applicable if there is more than one cluster in the SRP); 5) Some metadata that represents the *Authors* that created or modified an SRP, the *Sources* where the knowledge that induced to think about the convenience to create an SRP was extracted, the *Date* when an SRP was created or later modified, and the current *Version* of an SRP; and 6) Some information about the *Type* a Domain has, as well as its *Invariant*.

Table 4.3 shows the *Users Capacity* SRP, already introduced in Section 3.2, following the structure of the metamodel. This pattern will be used in a project in case the customer needs a system with a minimum number of users, so it was constructed from, among others, requirements Req-23 and Req-24 (see Table 4.2). As stated in the consequence of Statement 4, it is possible to apply the SRP several times in the same project, no matter the SRP cluster applied, providing that the applications are over different combinations of the *Subsystems* and *User Profiles* (entity types). This is a consequence of Statement 4 of the previous section, but for an ease of understanding it has also been materialized in the first rule of the behaviours of the SRP clusters.

In each application, one cluster is used depending on: the customer wants to define the users capacity per user profile (*Users Capacity by Profile* cluster), or to define the global capacity of users, i.e. without taking into account the different types of users in the system *(Global*

Users Capacity cluster). The two extended parts of both clusters are optionally used when it is required to state the growing of registered users, either by a specific number of users or by a percentage on the number of users. As the behaviour part states, the extended parts of the *Global Users Capacity* cluster can be used just once, and not together, per requirements cluster (i.e. for the same *subsystem*). In the case of the *Users Capacity by Profile* cluster, the behaviour states that the extended parts can be used more than once, and not together, per requirements cluster (i.e. for the same *subsystem*) if they are applied with disjoint values of *user profiles*. The behaviour also states that the two extended parts should use the same *user profiles* as the ones used in the fixed part's applications for the same *subsystem*.

Table 4.3 – Users Capacity SRP

SRP ***Users Capacity***	**Goal**	Supporting a required number of users in the system.		
	Keywords	Users, Records, Limit, Concurrent		
	Description	This SRP expresses the need for the system to support a required number of users registered in the system.		
	Question	Does the customer require different user capacities depending on the profiles of the users?		
	Authors	GESSI-LIST		
	Sources	Specialized literature, SRSs from LIST		
	Date	13 January 2016		
	Version	v.3		
	Entities	Subsystem, User profile		
	Relationships	- *Users Capacity* SRP relates-to *Concurrent Users Capacity* SRP - *Users Capacity by Profile* cluster (*Users Capacity* SRP) may require *Authorization* SRP - *userProfiles* parameter could be the same in and/or should be consistent with the applications of the following SRPs: *Authorization*, *Interface Learnability*, *Online Help*, *Users Capacity*		
SRP Cluster *Global Users Capacity*	**Description**	This SRP cluster expresses the general need for the system to support a given amount of registered users of any kind. It has extensions for expressing the allowed growing of registered users that should not interfere with system's behaviour.		
	Behaviour	- *Fixed Part*, *Growing of Registered Users by Number*, *Growing of Registered Users by Percentage*: cannot be applied more than once for the same *subsystem*. - *Growing of Registered Users by Number*, *Growing of Registered Users by Percentage*: cannot be applied together for the same *subsystem*.		
	Fixed Part	**Template**	*The %subsystem% shall be able to support %usersNumber% users.*	
		Parameter name	**Domain name**	**Domain type**
		usersNumber	*NumberOfUsers*	*Integer > 0, "any type of"*
		subsystem	*Subsystem*	*[System, Server system, Client sybsystem, ...]*
	Extended Part *Growing of Registered Users by Number*	**Template**	*The %subsystem% shall allow a growing of a minimum of %usersNumber1% users.*	
		Parameter name	**Domain name**	**Domain type**
		usersNumber1	*NumberOfUsers*	*Integer > 0*
		subsystem	*Subsystem*	*[System, Server system, Client sybsystem, ...]*
	Extended Part *Growing of Registered Users by Percentage*	**Template**	*The %subsystem% shall allow a growing of %volPercentage% on the number of users.*	
		Parameter name	**Domain name**	**Domain type**
		volPercentage	*VolumePercentage*	*Float > 0*
		subsystem	*Subsystem*	*[System, Server system, Client sybsystem, ...]*

SRP Cluster *Users Capacity by Profile*	**Description**	This SRP cluster expresses the general need for the system to support a given amount of registered users of certain profile. It has extensions for expressing the allowed growing of registered users of a certain profile that should not interfere with system.		
	Behaviour	- *Fixed Part*, *Growing of Registered Profile Users by Number*, *Growing of Registered Profile Users by Percentage*: can be applied more than once if they are applied with disjoint values of *userProfile* and *subsystem*. - *Growing of Registered Profile Users by Number*, *Growing of Registered Profile Users by Percentage*: cannot be applied together for the same value of *userProfile* and *subsystem*. - *Growing of Registered Profile Users by Number* and *Growing of Registered Profile Users by Percentage* applications should use a subset of the *userProfile* used in *Fixed Part's* applications for the same *subsystem*.		
	Fixed Part	**Template**	*The %subsystem% shall be able to support %usersNumber% of the %userProfile% profile.*	
		Parameter name	**Domain name**	**Domain type**
		usersNumber	*NumberOfUsers*	*Integer > 0, "any type of"*
		userProfile	*UserProfile*	*[Administrator, Technical, User, Author, ...]*
		subsystem	*Subsystem*	*[System, Server system, Client sybsystem, ...]*
	Extended Part *Growing of Registered Profile Users by Number*	**Template**	*The %subsystem% shall allow a growing of a minimum of %usersNumber1% users of the %userProfile% profile.*	
		Parameter name	**Domain name**	**Domain type**
		usersNumber1	*NumberOfUsers*	*Integer > 0*
		userProfile	*UserProfile*	*[Administrator, Technical, User, Author, ...]*
		subsystem	*Subsystem*	*[System, Server system, Client sybsystem, ...]*
	Extended Part *Growing of Registered Profile Users by Percentage*	**Template**	*The %subsystem% shall allow a growing of %volPercentage% users on the number of users of the %userProfile% profile.*	
		Parameter name	**Domain name**	**Domain type**
		volPercentage	*VolumePercentage*	*Float > 0*
		userProfile	*UserProfile*	*[Administrator, Technical, User, Author, ...]*
		subsystem	*Subsystem*	*[System, Server system, Client sybsystem, ...]*

4.3 Relationships among SRPs

The analysis of the relationships among requirements in SRSs is articulated in Subsection 4.3.1 through just one, but fundamental, statement supported by some observations that drove to a consequence. The representation of this consequence over the final metamodel is presented in Subsection 4.3.2.

4.3.1 Analysis of SRS Documents

Statement 8

> *A requirement is not an isolated unit of knowledge; instead, there are several types of relationships among the different requirements of an SRS document.*

Observation. The analysed SRSs show clearly these relationships and also the diversity of the relationships granularity:

- Relationships at cluster level. One example is among requirements referring to the *Stored Data Protection* concept and the ones aligned to the *Authorization* concept (see Table 4.4). It is quite clear that it is necessary to have the capability of giving authorizations to system users in order to be able to protect stored data from unauthorized users, so this relationship is of type *requires*. Other types of relationships found at this level are *conflicts* (the achievement of some requirements could hinder the achievement of other requirements), *supports* (the achievement of some requirements could help on the achievement of other requirements) and the most general one *relates-to* (the specification of some requirements could give interest to the specification of other requirements). After studying all the clusters in the SRS documents, the analysis shows that there are 377 relationships among clusters (see Figure 4.8). Experts believed that when all the clusters of two SRPs had the same relationships, the relationship could be considered as a relationship between the SRPs.
- Relationships at requirement level. For instance, Req-31 in Table 4.4 is related to the requirement Req-11 in Table 4.1, since if one is included in a project, it would be likely that the customer was interested in the other one, because both refer to the notification of certain system events to their users (relationship of the type *relates-to*). While analysing all the requirements in the SRS documents, 3 relationships of this type appear among individual requirements (see Figure 4.8). Although no other types of relationships at this level were found in the analysed SRSs, expert assessment revealed some other potential types that may appear in the future. Specifically, they thought *conflicts*, *relates-to* and *supports* types could be found at this level. The four types are defined as above.
- Relationships at project specific information level. This can be seen in Req-20 of Table 4.2 and Req-36 and Req-37 of Table 4.4, which express different constraints related to the profiles of the users. Req-20 defines the user profiles the system has to manage, while Req-36 and Req-37 specify the number of users of a specific user profile the system has to support. Because of that, it does not make sense that the profiles used in Req-36 and Req-37 do not appear in Req-20, i.e. the relationship is of type *subset* between the user profile (the project specific information) in Req-36 and Req-37 with respect to Req-20. Other types found at this level while analysing the SRS documents are *supports* (defined as above), *rule* (a formulae between the

specific information of two requirements) and *disjoint* (the specific information of two requirements does not have overlapping values) (see Figure 4.8 for the distribution per type of relationship). Other types that were not found in the analysis at this level, but that experts considered to appear in the future are: *conflicts*, *relates-to* and *requires*, which are defined as above.

- Relationships among different levels. This is illustrated in the requirements referring to the *Logs* concept and the ones aligned to the *Authentication* concept in SRS-1 (see Table 4.4): the use of *username* as project specific information while defining the information to log (specified in Req-33) requires having some authentication mechanism in the system (cluster composed by Req-34 and Req-35). In the analysed SRSs, there are three relationships of type *requires* among the levels project specific information and cluster (see Figure 4.8). No other types of relationships were found between these two levels, but expert assessment revealed the possibility of having relationships of the types *conflicts*, *relates-to* and *supports* in the future (defined as above). Apart from the relationship between these two levels, the analysis did not reveal relationships among other different levels; however, experts did not discard their appearance.

Existence of relationships among software requirements is a well-documented fact [155]. The NFR framework approach, proposed by Chung et al. [170], proposes a rich range of possibilities that facilitates even some automatic treatment (*Make, Break, Help, Hurt, Some+, Some-, Equal, Unknown*). Other approaches use intermediate artefacts to help in discovering, tracing and expanding those relationships, like the works of Franch and Carvallo [171] and Egyed and Grünbacher [172], which use quality models with this purpose. Moreover, Withall [9] and Moros et al. [168] give support to the existence of relationships among project specific information in requirements. Withall talks explicitly about "a requirement using information defined in another requirement", while Moros et al. state that relationships among requirements variability points exist.

Consequence. An abstract metaclass *Element* has been added to the metamodel, which has five heirs, one for each different SRP element: *SRP*, *SRP Cluster*, *Requirement Abstraction*, *Parameter* and *Domain*. Due to the variety of the granularity of the relationships found among these elements, there is an association from *Element* to *Element* with an association class

Relationship. As *Relationships* can be of different types, an attribute *Type* has also been incorporated (which can be *conflicts*, *relates-to*, *requires*, *supports*, *disjoint*, *rule* and *subset*). In order to better understand a *Relationship*, a field *Description* has also been added to it. ■

Table 4.4 – Excerpt of requirements found in the analysed SRS documents (3)

Req. Id	Concept	Requirement	Project Entity	SRS Id
Req-25	Stored data protection	**Personal data** kept in the **system** shall not be accessible to unauthorized system user profiles or to people attempting to access from outside the system	Whole system	**SRS-5**
Req-26	Stored data protection	**Personal data** must be encrypted in the database.	Whole system	**SRS-5**
Req-27	Authorization	The **system** shall control user access rights to all kind of resources.	Whole system	**SRS-5**
Req-28	Authorization	The **system** must allow the restriction of data access to certain people depending on the type of data.	Whole system	**SRS-5**
Req-29	Authorization	The **system** must allow the restriction of operations to some people.	Whole system	**SRS-5**
Req-30	Authorization	The **system** can manage the following profiles: **Manager, Author and Validator**.	Whole system	**SRS-5**
Req-31	Failures	In case of error or failure, the user and technical team must be informed in an understandable and unambiguous way to the existence and nature of the error occurred.	Whole system	**SRS-3**
Req-32	Logs	The **system** should maintain an historical record of these transactions.	Whole system	**SRS-1**
Req-33	Logs	The essential data to trace are: **username, date, and accessed or modified data.**	Whole system	**SRS-1**
Req-34	Authentication	The **system** should authenticate the users.	Whole system	**SRS-1**
Req-35	Authentication	Authentication will be via **the Windows login**.	Whole system	**SRS-1**
Req-36	Users Capacity	The **solution** should permit **250** users of the profile **Secretariat**.	Whole System, User Profile	**SRS-2**
Req-37	Users Capacity	The **solution** should permit **50** users of the profile **Directors**.	Whole system, User Profile	**SRS-2**

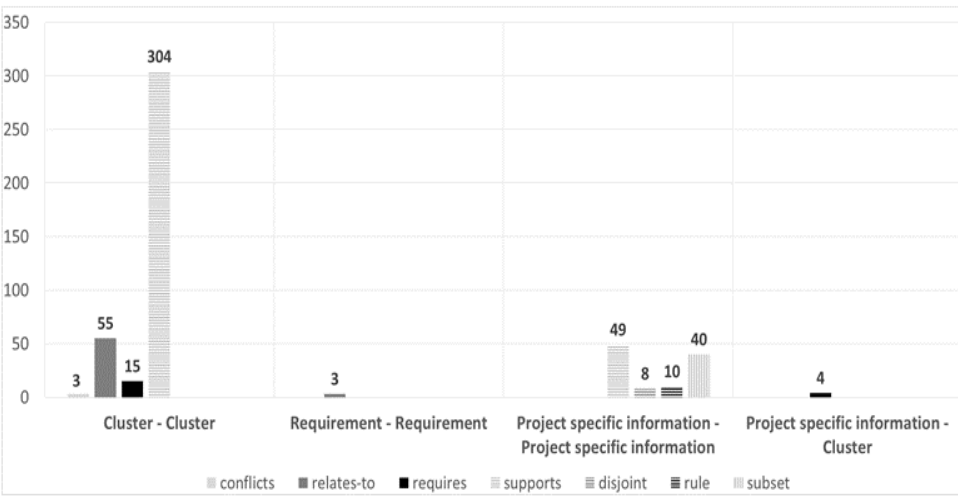


Figure 4.8 – Relationships found in SRS documents distributed per granularity and type

4.3.2 Metamodel of SRP Relationships

Figure 4.9 shows the final metamodel that represents the SRP relationships derived from the consequence of the statement. Integrity constraints play a fundamental role in this part of the metamodel. This is why the following constraints need to be taken into account when defining relationships among SRP elements:

- A relationship declared at one level among two elements A and B cannot contradict a relationship stated at any other level among two different elements of A and B.
- There cannot exist cycles in the relationships. For instance, if element A requires element B, and element B requires element C, element C cannot require element A.
- As a consequence of Statement 8, not all types of relationships are applicable to all the possible relationships. In particular, experts agree that the types *disjoint*, *rule* and *subset* are only applicable when both elements of a relationship are *Parameters*.

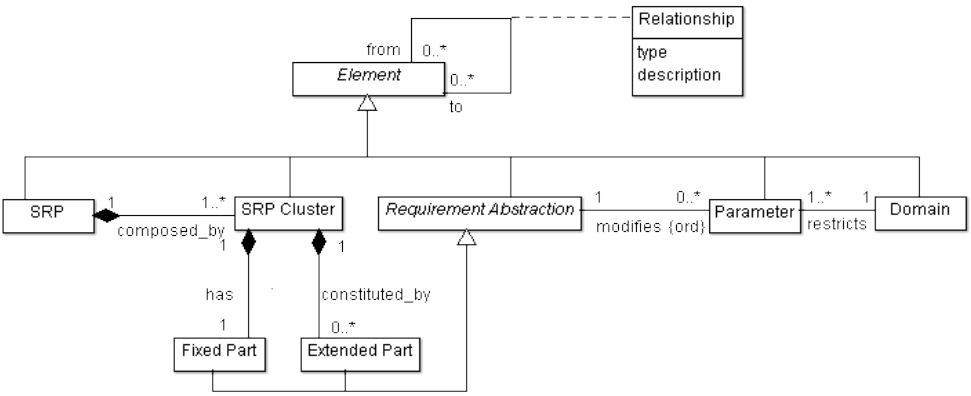


Figure 4.9 – Metamodel: SRP relationships

Figure 4.10 shows an excerpt of the relationships found among NF-SRPs. The *Users Capacity* SRP has all its relationships represented in the figure: it relates to the *Concurrent Users Capacity* SRP (as the capacity of users should be consistent with the number of concurrent users the system shall support with good performance); its *User Capacity by Profile* cluster requires the use of the *Authorization* SRP (as it is necessary that the system manage profiles in order to be able to differentiate the capacity of users of a certain profile); and the values of its parameter *userProfile* shall be consistent with the ones defined in the *userProfile* parameters of the *Authorization* SRP, the *Interface Learnability* SRP, and the *Online Help* SRP (as it does not make sense to define the same user profile with different names in different requirements).

Other relationships depicted in the figure are, for instance: the *Platform* SRP and the *Online Help* SRP should support each other (because the requirements specified using both SRPs are defining technologies, and it is necessary that these technologies are consistent), and the use of the value *username* of the domain associated to the *logInformationType* parameter of the *Logs* SRP requires the use of the *Authentication* SRP (since it is not possible to log the username of a user if it is not authenticated in the system).

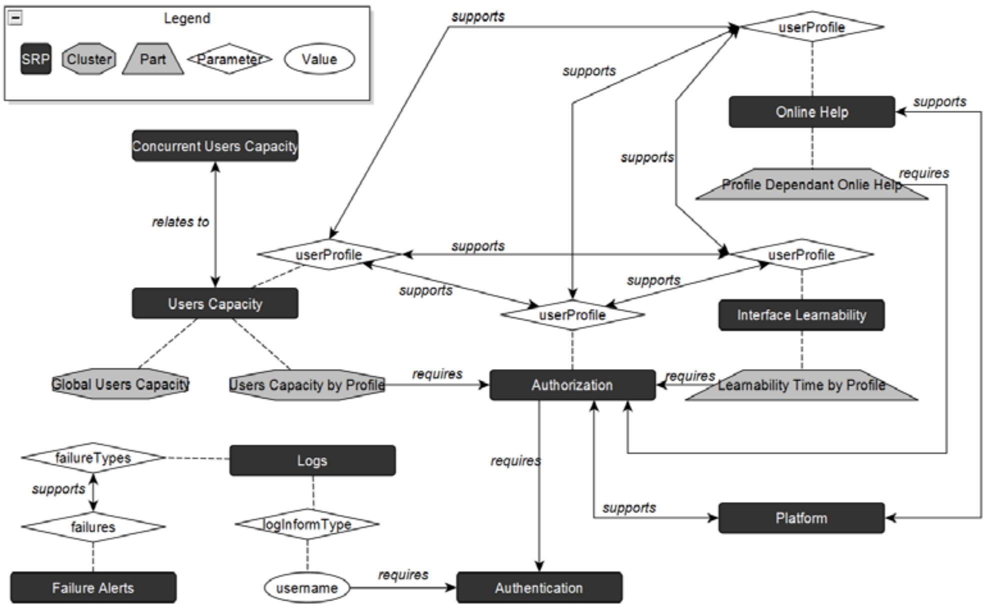


Figure 4.10 – Excerpt of the relationships found among NF-SRPs

4.4 Classification of SRPs

In SRS documents, requirements are classified to give structure to the documents and to help browsing them. Subsection 4.4.1 presents statements and observations on the classification of the requirements in the analysed SRS documents and their consequences. Subsection 4.4.2 presents their representation in the metamodel for the SRP classification.

4.4.1 Analysis of SRS Documents

Statement 9

> *Requirements appear classified in different ways in different SRS documents.*

Observation. All the requirements in the 6 analysed SRS documents are classified. Although all the SRS documents have the same classification, GESSI members are used to apply a different classification of NFRs. Thus, the existence of different classifications was observed from the experience of experts, IT consultants and GESSI members. In the literature, Glinz [4] also states that there are different proposals of classification of NFRs.

Therefore, in this case, the experts' knowledge and literature review prevailed over the SRS documents observation.

Consequence. There exists a class *Classification Schema* in the metamodel that represents different views of the SRP catalogue. The classification schemas offer different alternative classifications or structuring schemas of the catalogue. ■

Statement 10

> *The classification of requirements in an SRS document is structured as a hierarchy of different classifiers.*

Observation. The analysed SRS documents present a flat hierarchy of classifiers for the organization of NFRs, below a first level with three big categories corresponding to the main division among FRs, NFRs and Call-for-Tender requirements (these Call-for-Tender requirements are the ones corresponding to NTRs). Specifically, NFRs are generally structured in 9 big sections in the SRSs (*Databases*, *Data migration*, *Ergonomics*, *Infrastructure*, *Installation process*, *Interoperability*, *Maintenance*, *Performance*, and *Security*), and few differences existed among the classifications of the different documents (see Figure 4.11). There is only one shorter SRS that does not present requirements in four of these classifiers. In this observation, the influence of experts' advice and literature review also prevailed over the observation of just SRS documents, since the idea obtained from these sources was to allow hierarchical classifications to give more flexibility to the classification schemas.

In the literature, it is possible to find classifications based on a flat list of classifiers, e.g. the IEEE 830 standard [138], and other ones that are structured as a hierarchy of classifiers, as the standards ISO/IEC 25010 [132] and ISO/IEC 9126 [137], and Davis' et al. work [173]. However, even flat lists of classifiers usually have an upper level of classifiers for differentiating among types of requirements (FRs, NFRs, and NTRs).

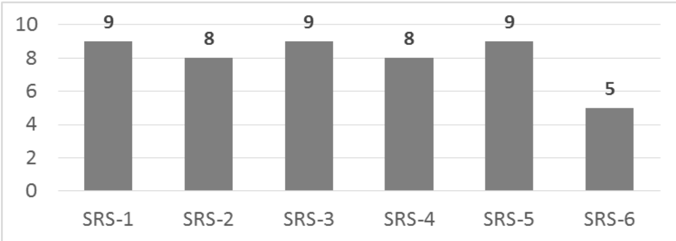


Figure 4.11 – Number of classifiers per SRS document

Consequence. A class *Root* in the metamodel represents the root classifiers in a *Classification Schema*. *Roots* are *Compound Classifiers*, which may be decomposed in *Compound Classifiers* (i.e. classifiers containing other classifiers) or in *Basic Classifiers* (i.e. the ones indexing *SRPs*). ■

Statement 11

Inside a classifier of an SRS document, there are requirements related to different concepts.

Observation. Out of the 48 non-functional classifiers found in the analysed SRS documents, 44 classifiers contain requirements related to different concepts in the same SRS (see Figure 4.12); only 4 classifiers contain requirements related to the same concept.

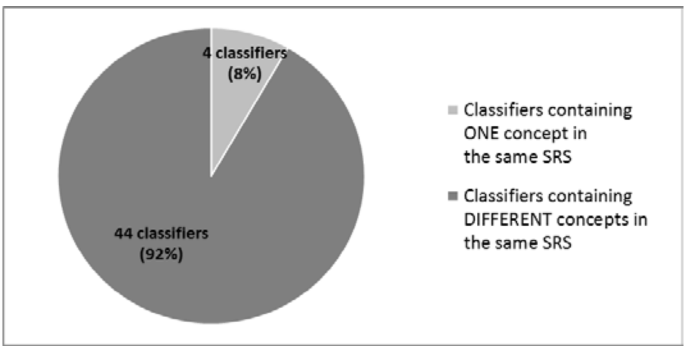


Figure 4.12 – Classifiers containing more than one concept in the same SRS

In the literature, the concept of classifiers containing requirements that constraint different aspects (i.e. requirements related to different concepts) is widely used. Books about RE support this idea when talking about requirements classifications (e.g. Pohl and Rupp [6], and Hull et al. [1]). Proposals for automatizing the process of classifying requirements also take into account this fact, and allow to place different requirements in the same classifier (e.g. Cleland-Huang et al. [174] and Minhas et al. [175]). What is more, Withall proposes in his book about for RE patterns [9] the term *domain* (what it is called here classifier) to group patterns that share a common theme.

Consequence. Each *Basic Classifier* may have associated several *SRPs*. ■

Statement 12

> *Requirements in an SRS document could be classified according to different classifiers of the same document.*

Observation. In the SRS documents, NFRs are classified below just one classifier of the document. However, after studying them, it is possible to see that there are a number of requirements that could also be included in another classifier since they could be considered to be related to both of them. For instance, Req-31 of Table 4.4 is related to the *Ergonomics of the system*, but is also related to its *Maintenance*, so it could be placed under both classifiers. This situation happens in 20 NFRs of the analysed SRS documents out of 417 NFRs. Although the ratio is low, it was thought that it was better to make classification schemas as general as possible and that possibility was incorporated, so if the situation appears, even if it is just in a small part of the requirements, it could be applied.

Other works have also considered the possibility of classifying requirements under different classifiers. Hull et al. [1] states that requirements can be given primary and secondary classifications, where the primary classification is the one related to the position of the requirement in the SRS (i.e. the section or classifier where the requirement is placed), and the secondary classifications correspond to other classifiers related to the requirement. Cleland-Huang et al. [174], in their automatic classification process of requirements, also considers the possibility of classifying the same requirement in more than one classifier. The same is considered in Minhas' et al. [175] requirements classification process for NFRs.

Consequence. One *SRP* may be related to several *Basic Classifiers*. ■

4.4.2 Metamodel of SRP Classification Schemas

Figure 4.13 shows the final metamodel that represents the structure of the SRP classification schemas derived from the consequences of each statement.

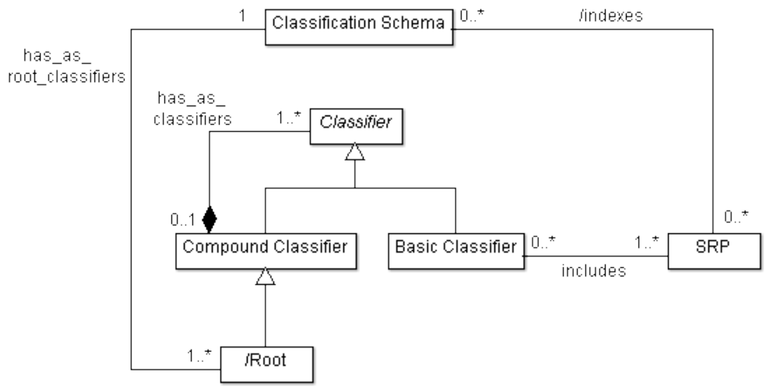


Figure 4.13 – Metamodel: SRP classification schemas

Table 4.5 – Excerpt of the ISO/IEC 25010 based classification schema with NF-SRPs classified

ISO/IEC 25010 Classifiers		SRP
Performance Efficiency	Time Behaviour	Interface Load Time
	Resource Utilization	-
	Capacity	Concurrent Users Capacity
		Data Capacity
		Users Capacity
Compatibility	Co-existence	-
	Interoperability	Data Exchange
		Interoperability with External Systems
Usability	Appropriateness Recognisability	-
	Learnability	Interface Learnability
		Online Help (r)
	Operability	*Failure Alerts (r)*
		*Installation Procedures (r)**
		Online Help (r)
		Recovery Procedures
		*Update Procedures (r)**
	User Error Protection	-
	User Interface Aesthetics	-
	Accessibility	Interface Language
		Interface Type
Reliability	Maturity	*Failure Alerts (r)*
	Availability	Availability
		Downtime
		Uptime
	Fault Tolerance	Alternative Data Storage
	Recoverability	Backups
		Logs

* These SRPs are indexed by other classifiers not shown in this table. For a complete version of the classification schema see Chapter 7

Table 4.5 shows an excerpt of the ISO/IEC 25010 [132] based classification schema with some NF-SRPs already classified (the complete version of the schema can be found in Chapter 7). In this excerpt there are 4 root classifiers (corresponding to the first level of classifiers). Regarding the second level of classifiers, 10 of them are basic classifiers (the ones that have SRPs classified) and 5 of them are compound classifiers (the ones not having SRPs classified). In the future, the compound classifiers might get SRPs classified on them (and then their type will change to basic classifier), or they might be decomposed into more compound and basic classifiers. It is possible to see that the majority of SRPs are classified in just one basic classifier (e.g. the *Users Capacity* SRP, which is classified under *Performance Efficiency* → *Capacity*). However, there are some SRPs that are classified in different basic classifiers (marked in italics and with *(r)* in the table). For instance, the *Failure Alerts* SRP is classified under *Usability* → *Operability* and *Reliability* → *Maturity* because the requirements coming from applying this SRP can be included either in the *Operability* section or in the *Maturity* section of an SRS.

CHAPTER 5 Use of an SRP Catalogue

5.1 Introduction

When an SRP catalogue is used during the elicitation and specification of requirements, the requirements analyst identifies and applies the SRPs in the catalogue that are suitable for the IT project of the customer. This chapter presents the different alternatives proposed in the PABRE framework for guiding the use of the SRP catalogue. As an overview, the common aspects of the alternatives are presented here adopting as a guide the dimensions of requirements reuse described by Wiegers and Beatty [26] (already presented in Subsection 1.1.2).

The *extent of requirements reuse* are SRPs. More specifically, what is reused in the PABRE framework are parts of an SRP that together solve the problem stated in the goal of the SRP.

The *reuse mechanism* corresponds to the process of selecting and applying SRPs. The PABRE framework incorporates different alternatives to use the SRP catalogue during the requirements elicitation and specification phases of IT systems. During this use, requirements analysts select from the catalogue an SRP that applies to the particular IT project at hand, and convert it into requirements; this cycle is performed in an iterative way (an SRP is selected and afterwards applied) until requirements analysts get all the requirements that finally configure the SRS (Figure 5.1). Therefore, all the reuse methods proposed become a process of searching, picking-up and applying SRPs from the SRP catalogue. As thought in the framework, the process is performed through interviews between a requirements analyst and a customer, and all decisions are agreed between the both of them.

The *extent of modifications* corresponds to the customizations that are allowed while using SRPs. Although it is expected that requirements will come directly from the application of SRPs, other situations may occur: 1) An SRP has to be slightly modified when becoming a requirement, probably because small details about the wording of the requirement have to be adapted to the context of the customer; 2) Some requirement cannot be created as an SRP application, either because the requirement is very specific of the project, or because the SRP catalogue is still not complete enough.

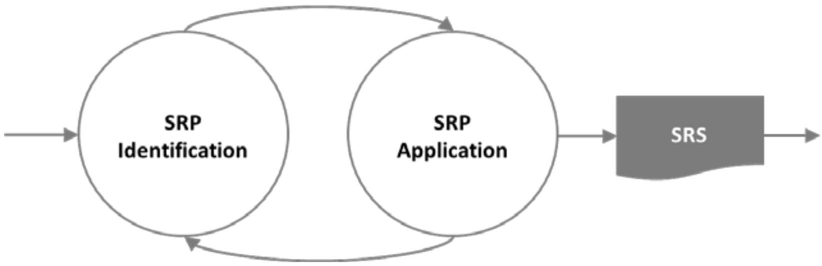


Figure 5.1 – Overview of the PABRE use methods

Section 5.2 presents the different alternatives to guide the requirements analyst for exploring the SRP catalogue and identifying the suitable SRPs for a specific IT project. Section 5.3 provides further details about how to apply an SRP when it has been already selected, and the special situations that may arise when doing that.

5.2 Identification of SRPs

There are different approaches for the identification of SRPs depending on the mechanism used. The approach based on browsing and searching the catalogue is described in Subsection 5.2.1, and the rest of approaches are presented in Subsection 5.2.2

5.2.1 Browsing/Searching the SRP Catalogue

The browsing approach is based on the use of the SRP catalogue using one of the classification schemas of the catalogue and/or the relationships defined among SRPs in the catalogue. The browsing of the catalogue is optionally complemented by a search approach, which allows identifying the SRPs that have in their definition the terms used in the search. Once an SRP whose goal meets some customer problem or need is identified, the SRP clusters and subsequently the parts more suitable to achieve the goal in the system under development are considered. Therefore, the main steps for the SRP identification are: *SRP Exploration*, *Cluster Exploration* and *Part Exploration*. The flow of these steps is specified in detail in Figure 5.2, and described and illustrated in the following through an example, namely the *Users Capacity* SRP (already introduced in Chapter 4 in Table 4.3).

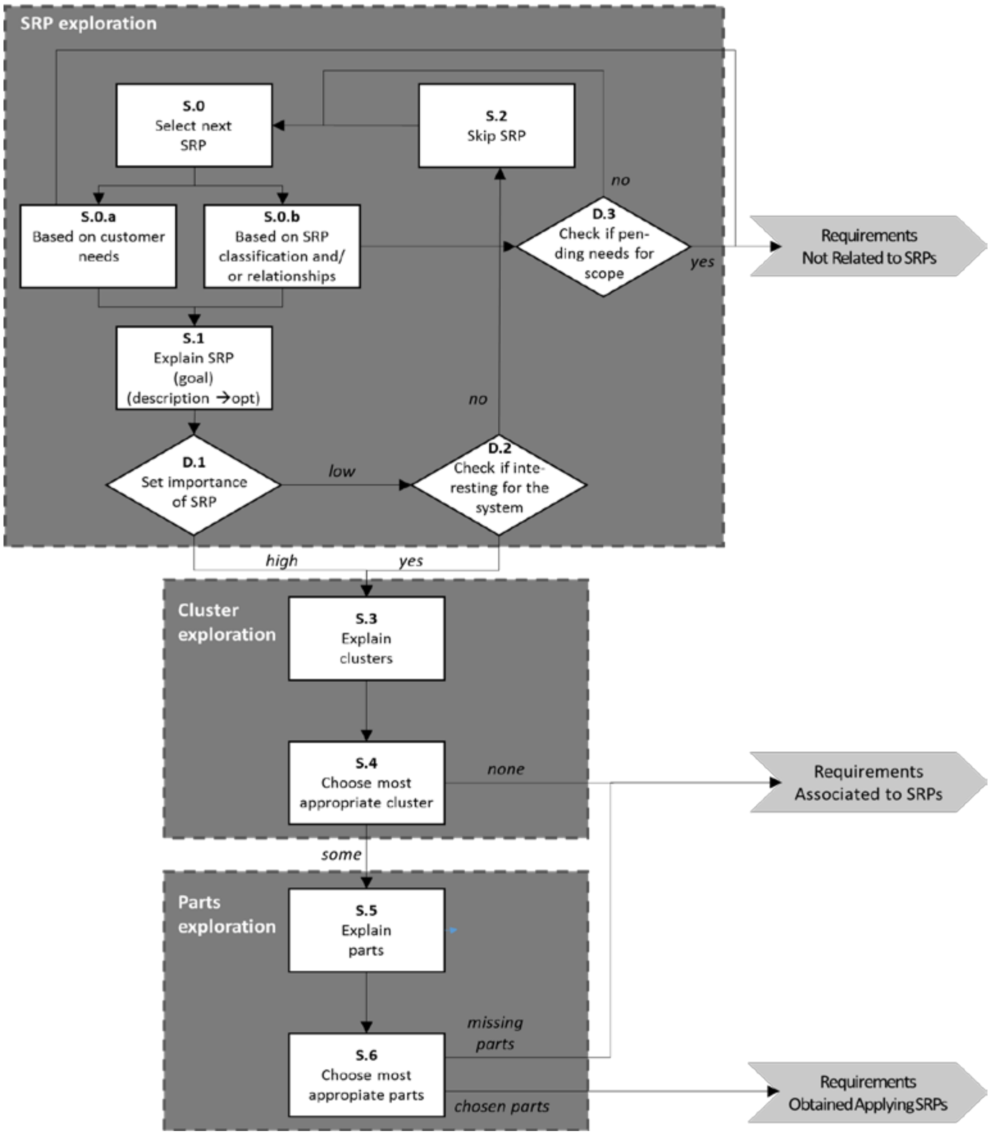


Figure 5.2 – Activities followed during the SRP identification with PABRE

SRP Exploration

Select next SRP (S.0). The aim of this step is to look for SRPs that are suitable for the customer problems or needs. The requirements elicitation meeting among the requirements analyst and the customer may have different flows (step S.0.a and S.0.b). Sometimes it may be guided by the explanation of a problem or need given by the customer (step S.0.a), and sometimes by suggestions of the requirements analyst based on the classification schema and/or SRP relationships (step S.0.b).

Based on the customer needs (S.0.a). In this case, the customer explains his/her interest in a specific requirement, or that s/he has a specific problem or need. Let us assume that the customer is interested in stating the number of users that have to be able to use the system under development. Either the knowledge that the requirements analyst has about the catalogue or a search in it may uncover the existence of the *Users Capacity* SRP, which has as goal *Supporting a required number of users in the system*. The attributes of an SRP that may be used for the search are: name, goal, keywords and description. In case there is not any SRP in the catalogue that may help to solve the need or problem of the customer, it means that there is a requirement not covered by any SRP that needs to be included in the SRS (*Requirements not related to SRPs* in Figure 5.2).

Based on the SRP classification/relationships (S.0.b). When the requirements analyst suggests the next goal for the system under development, s/he use either the next SRP according to the selected classification schema or the relationships of the previous identified SRP. Let us assume that the last identified SRP is *Users Capacity*. In the case of the classification schema, the requirements analyst may propose the application of the *Data Capacity* SRP, since both patterns are classified under the same subcategory of the classification schema (see the classification of both SRPs in Figure 5.3). Otherwise, in case of using relationships, the requirements analyst may propose the *Authorization* SRP that allows defining the user profiles that the system under development has to have, since there is a relationship among the *Users Capacity per Profile* cluster of the *Users Capacity* SRP and the *Authorization* SRP.

Check if pending needs for scope (D.3). It is important to remark that, when the requirements analyst follows a classification schema, and the previous identified SRP is the last one bound to the current scope (i.e. classifier), before changing the scope, s/he will ask to the customer if there are still some needs related to the scope that have not been covered yet. If there exists any requirement that fits this condition, it means that there is a requirement not covered by any SRP that needs to be included in the SRS (*Requirements not related to SRPs* in in Figure 5.2).

Explain the SRP goal (S.1). Whatever is the case, after selecting the next SRP, the requirements analyst explains the goal of the SRP to the customer. The SRP goal is the key knowledge asset to decide the adequacy of an SRP in the given project. If further information is needed, the description of the SRP can also be explained.

Set the importance of SRP (D.1). Based on this explanation, the requirements analyst asks the customer to define the importance of the pattern.

Check if interesting for the system (D.2). If the customer considers that the SRP is not important for the system, the requirements analyst determines with the customer whether the SRP matches a need or solves a problem of the customer.

Skip SRP (S.2). In case the customer and the requirements analyst arrive to the conclusion that the SRP is not relevant for the system under development, the SRP is skipped without processing its elements entirely. Then the requirements analyst proceeds to the next SRP according to the customer needs, classification schema or relationships.

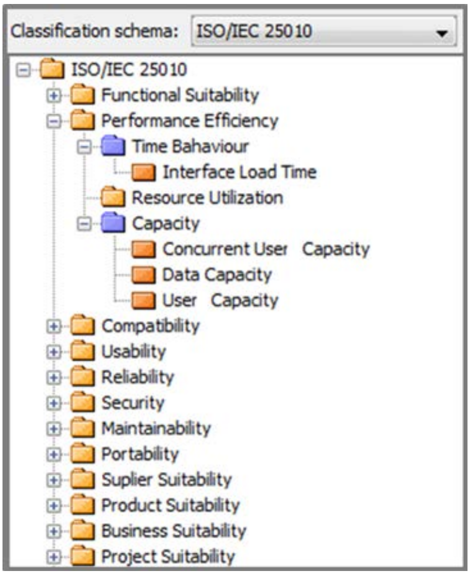


Figure 5.3 – Excerpt of the ISO/IEC 25010 classification schema

Cluster Exploration

Explain the clusters (S.3). Assuming the customer and the requirements analyst have chosen an SRP (because either it is important for the system, it matches a need of the customer, or it solves one of the customer problems), the requirements analyst explains the cluster(s) of the SRP (i.e. the different forms to achieve the goal of the SRP). To do so, the requirements analyst uses the question that helps to decide among clusters (only if the SRP has more than one cluster) and the cluster(s)' descriptions. In the *Users Capacity* SRP, the question is *Does the customer require different types of alerts depending on the types of failures?*, and the goal of the SRP can be attained by defining the users capacity depending on the user profiles (*Users Capacity by Profile* cluster), or by defining the global capacity of users, i.e. without taking into account the different user types (*Global Users Capacity* cluster).

Choose the most appropriate cluster (S.4). The customer, jointly with the requirements analyst, chooses the cluster that better suits the needs for the system under development taking into account the customer context. In the example, the *Users Capacity by Profile* cluster is selected in case the customer wants to fix the number of users by each role of user. It may happen that no suitable cluster is found. In this case, the requirements analyst needs to create one or more requirements to satisfy the goal (*Requirements associated to SRPs* in Figure 5.2). These new requirements are associated to the SRP, since they solve the goal defined for the SRP but they do not come from its application.

Parts Exploration

Explain the parts (S.5). Assuming the customer and the requirements analyst have chosen an SRP and one of its clusters, the requirements analyst explains the different parts composing the chosen cluster (i.e. the fixed and extended parts). If it is necessary, the requirements analyst skims over the parameters of each part and gives examples of possible values in order to facilitate their understanding.

Choose the extended parts (S.6). Jointly with the requirements analyst, the customer chooses the most convenient parts to achieve the SRP goal according to the customer context. As explained in the SRP structure (Section 4.2), the fixed part should always be selected when a cluster is chosen. In addition, the customer may choose several extended parts for a specific cluster, or even one extended part could be applied more than once with different assignments of values to parameters (all that is explained in the behaviour of the cluster). Following the example, suppose the customer chooses, from the *Users Capacity by Profile* cluster, the fixed part (as it always has to be chosen) and the extended part *Growing of Registered Profile Users by Number*, which deals with the growth of users by number. Eventually, some extended part not existing in the catalogue may be needed; again, it becomes necessary to elicit the missing needs separately (*Requirements associated to SRPs* in Figure 5.2). Once the parts of an SRP cluster have been selected, the new requirements are created from the application of these parts (*Requirements obtained from applying SRPs* in Figure 5.2).

5.2.2 Alternatives for Identifying SRPs

This subsection presents alternative approaches for identifying SRPs. These alternatives are less heavy for requirements analysts that are discovering the SRP catalogue or for requirements analysts that are more used to collect requirements in a less prescriptive manner. They also provide more transparency to the use of the SRP catalogue than the alternative presented in the previous subsection. It is worth remarking that these options are not completely exclusive, and they can be complementary one to each other.

Pre-selection of SRPs. This approach consists in a step that can be added prior meeting the customer to elicit the requirements. The idea is to pre-select SRPs on the basis of the available information regarding the current IT infrastructure and IT strategy of the customer. These pieces of information are usually collected before requirements elicitation and specification interviews take place. After this pre-selection of SRPs, the requirements analyst needs only to confirm his/her SRP analysis with the customer. However, this may introduce bias since requirements are no more elicited from the customer but deducted by the requirements analyst. This is why it is important to ask to the customer, after the pre-selected SRPs have been explored, if s/he believes some need is not being addressed.

SRPs as a checklist. The second approach consists in a task that can be added at the end of the meeting with the customer (where requirements have not been necessarily elicited and specified using SRPs), using the SRP catalogue as a checklist to validate that no requirement has been left out. The main goal is to include requirements that are usually forgotten, e.g. some non-functional ones. With that purpose, the requirements analysts skims the list of SRPs, and if s/he identifies some requirement that could have been forgotten, they are validated with the customer and included in the SRS.

SRP recommender. The last approach consists on using the SRP catalogue in a more transparent manner. The idea is that while the requirements analyst writes down requirements (which are not elicited using the catalogue) it may exists an automatic system that recommends SRPs, or parts of SRPs, with a similar goal to what the requirements analyst is typing (if such SRP or part exists in the SRP catalogue). The recommended items may be identified using the words written by the requirements analyst and the words in the SRP definition that correspond to the sections: names (of the SRP and each one of its clusters, parts, parameters and domains), goal (of the SRP), keywords (of the SRP), descriptions (of the SRP and each one of its clusters), templates (of each one of the parts), and possible domain values (of each one of the parameters). This process needs to be supported by a tool that implements recommendation and natural language matching techniques (e.g. the ones proposed in the Natural Language Toolkit[17] and the Stanford Pattern-Based Information Extraction and Diagnostics[18]). A student is currently developing a small prototype of the recommender using the Meteor[19] approach.

5.3 Application of SRP

In the process described in the previous section for identifying requirements using the PABRE method (Figure 5.2), requirements can be extracted in three different ways (Figure 5.4). The steps followed in the three situations are presented in Figure 5.5. Each situation is described below, and it is illustrated following the example already introduced in the previous section, based on the *Users Capacity* SRP (Table 4.3 of Chapter 4).

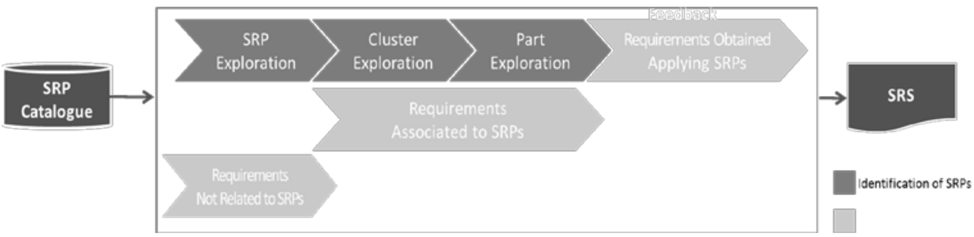


Figure 5.4 – Requirements extraction in PABRE

Requirements Obtained from Applying SRPs

The requirements are extracted by applying the selected parts of the identified SRPs.

Explain detailed information for parameters (if any) (S.7). For the chosen parts, the requirements analyst gives more details about the parameters of the parts (e.g. details on possible invariants, dependencies to/from other parameters, etc.) and explains the exhaustive list of values for each parameter. In the example, let us assume that during the SRP identification the customer has chosen the fixed part and the extended part *Growing of*

[17] http://www.nltk.org/
[18] http://nlp.stanford.edu/software/patternslearning.shtml
[19] http://www.cs.cmu.edu/~alavie/METEOR/

Registered Profile Users by Number of the *Users Capacity by Profile* cluster. These parts have three parameters corresponding to the subsystem, the number of users and the user profiles. Taking into account the parameters, the requirements analyst explains to the customer that it is necessary to fix the number of users that the system will have depending on the user profile, as well as the number of new users the system will allow per user profile. In case the system has several subsystems, this data can be established for each subsystem the customer wants (as stated in the behaviour of the cluster). If some parts do not contain parameters, this step is not necessary for these parts.

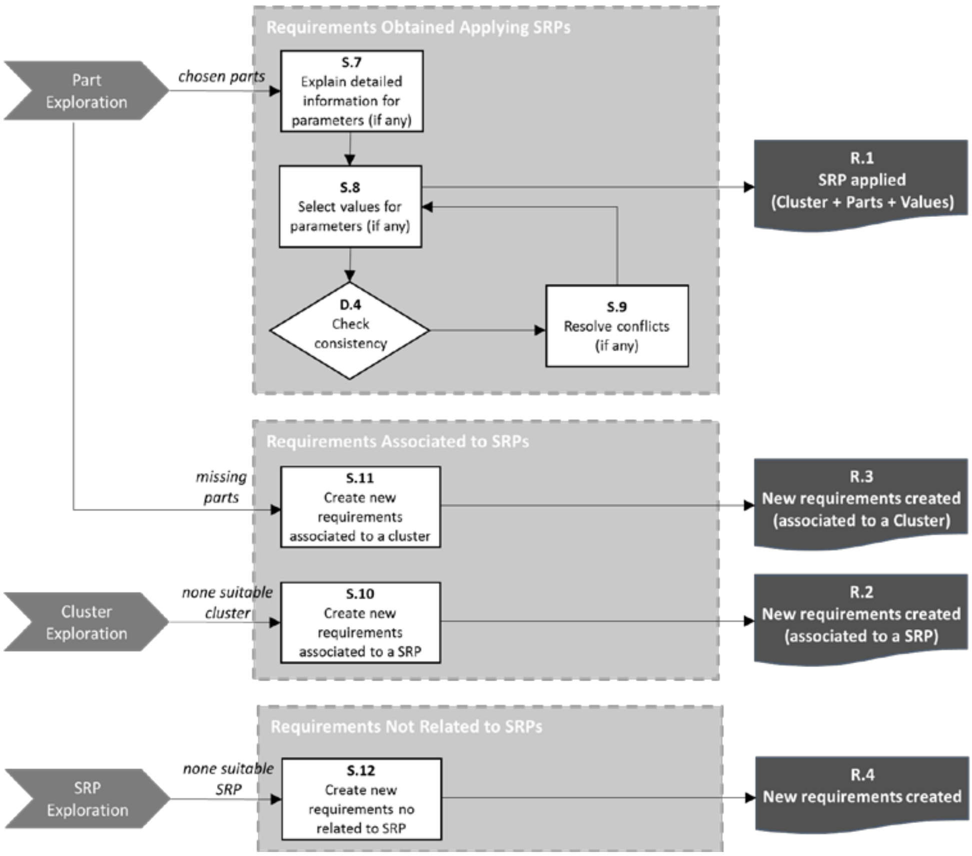


Figure 5.5 – Activities followed during the requirements extraction with PABRE

Select the values for each parameter (if any) (S.8). The customer chooses the values for the parameters, and agrees them with the requirements analyst. The requirements are extracted by applying the template of the selected parts with the parameters' values that have been agreed. At this point, it is possible to add a new value for the parameter if a proper value for the project at hand is not found in the SRP catalogue. Following the example, let us assume that the system under development does not has subsystems (i.e. only a system is implemented) and that the customer wants a system supporting 5,000 end-users and 5 administrators, and to allow a growth of a minimum of 15 administrators. The values of the parameters of the selected parts are filled according to this information (scenario M in Figure 5.6). It may happen that the requirements analyst or the customer does not like the concrete writing of a part. In that case, it can be changed during its application (alternative scenario C in Figure 5.6, where the customer prefers to state that the growth should be of "at least" some quantity rather than "a minimum of" a quantity). If some parts do not contain parameters, this step S.8 is not necessary for them.

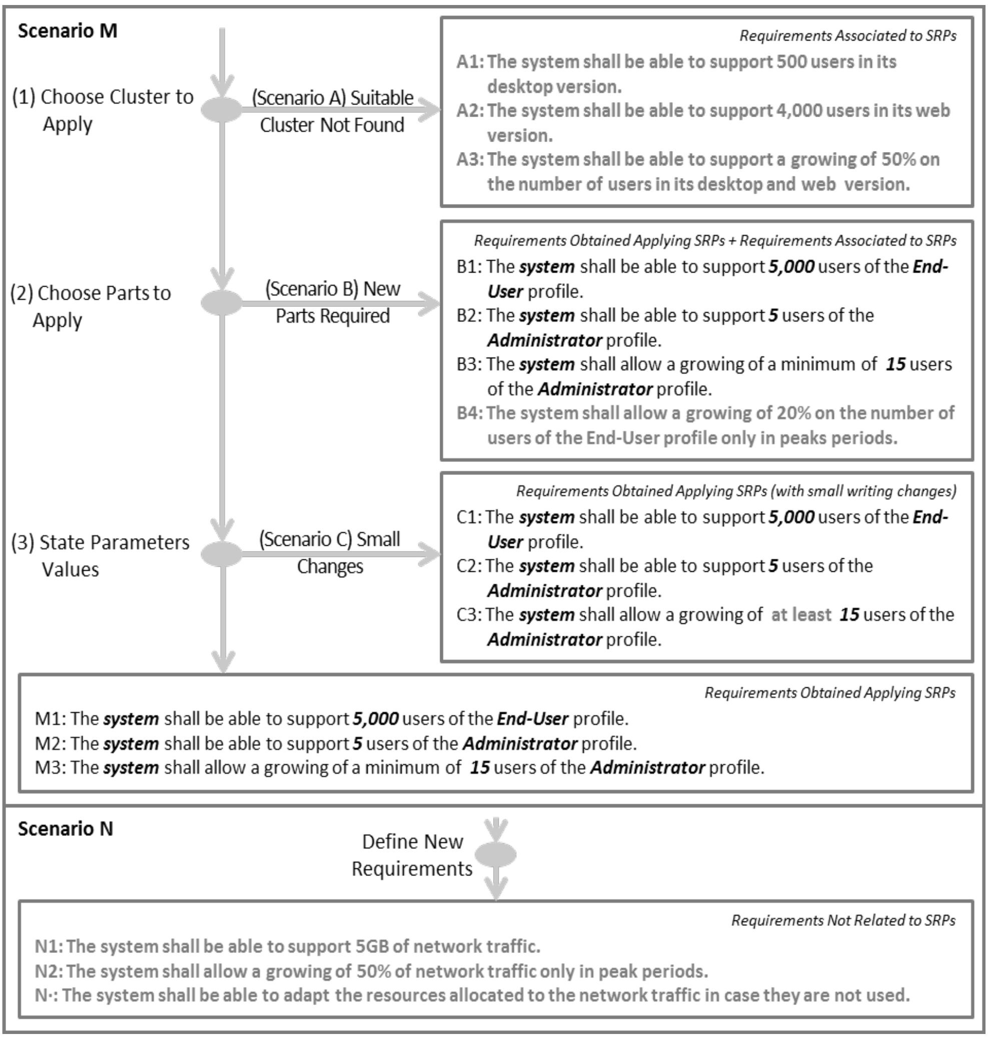


Figure 5.6 – Sample of requirements extracted with PABRE using the Users Capacity SRP

Check consistency (D.4). During the choice of values, the relationships with other SRPs and among the different applied parts are checked in order to verify the consistency between parameters and even between SRPs, i.e. all the rules stated in the behaviour of the selected cluster and the relationships of the SRPs are checked. In the case of the scenario M in the example, where the fixed part and extended part *Growing of Registered Profile Users by Number* of the cluster *Users Capacity by Profile* are applied, the behaviours lead to check if:

- The fixed part, the extended part *Growing of Registered Profile Users by Number*, and the extended part *Growing of Registered Profile Users by Percentage* of the cluster are not applied more than once for the same values of *userProfile* and *subsystem*.
- The extended part *Growing of Registered Profile Users by Number* and the extended part *Growing of Registered Profile Users by Percentage* are not applied together for the same value of *userProfile* and *subsystem*.
- The extended part *Growing of Registered Profile Users by Number* and extended part *Growing of Registered Profile Users by Percentage* applications use the same *userProfile* as the ones used in the fixed part's applications for the same *subsystem*.

It is easy to see that the three behaviours are fulfilled by the parts' applications of scenario M. As for relationships, three more statements need to be checked (see Section 3.2 for a detailed explanation of these relationships):

- The requirements obtained from the application of, or that are associated with, the *Concurrent Users Capacity* SRP are consistent with respect to the ones obtained from the application of, or that are associated with, the *Users Capacity* SRP.
- If *Users Capacity per Profile* cluster of the *Users Capacity* SRP is used, the *Authorization* SRP has been or will be used.
- In the requirements obtained from, or that are associated with, the SRP *Users Capacity*, *Authorization*, *Interface Learnability* and *Online Help*, the values of the parameters *userProfile* are consistent among all the requirements.

Let us assume that the *Authorization* SRP has been already applied, so the second rule is fulfilled, and that the user profiles stated in the requirements obtained from applying it are consistent with the ones stated in the scenario M, so the checking of the third rule is also satisfactory (assuming no requirements are obtained from, neither associated to, the SRPs *Interface Learnability* and *Online Help*). Let us also assume that there is a requirement obtained from the application of the *Concurrent Users Capacity* SRP stating *The system shall be able to support 1,000,000 users accessing concurrently ensuring acceptable system performance*. While checking the first rule, the requirements analyst notices that the customer is asking for 1,000,000 users accessing concurrently, while the system supports only 5,000 end-users. As the number of concurrent users is much bigger than the number of supported end-users, the checking of the first rule leads to a conflict.

Resolve conflicts (if any) (S.9). When the requirements analysts detects a conflict or an inconsistency, s/he warns the customer and they try to solve the conflict. Conflict resolution may not be straightforward and may even force to reconsider requirements already agreed. In the conflict example presented in D.4, where the number of concurrent users is much bigger than the number of supported end-users, the requirements analyst agrees with the customer that the requirement obtained from the application of the *Concurrent Users Capacity* SRP is incorrect. Therefore, the requirement is revisited again, and the number of concurrent users is modified to 2,000, giving place to the requirement *The system shall be able to support 2,000 users accessing concurrently ensuring acceptable system performance*.

Requirements Associated to SRPs

Sometimes, it may happen that the goal of an SRP is interesting for the project at hand, but that some requirements are missing in the SRP to fulfil the needs as the customer wants. The following situations requiring the creation of requirements associated to an SRP, but that are not coming from its application, have been identified.

Create new requirements associated to an SRP (S.10). This step is necessary in case the goal of an SRP has been considered as relevant for the system under development, but none of the SRP clusters fits well the needs of the customer. The requirements analyst will work with the customer a good way of expressing the form to reach the goal of the SRP that matches the needs of the customer. In the *Users Capacity* SRP used as example, there are two clusters depending on whether the customer wants different numbers of users according to the different user profiles or s/he is just interested in stating the global users capacity of the system. However, it may happen that the customer is not interested in any of the proposed solutions, and for instance s/he is interested in stating the users capacity for each one of the platforms of the system (alternative scenario A in Figure 5.6).

Create new requirements associated to a cluster (S.11). This step is necessary in case a cluster has been chosen but a requirement not captured by any existing extended part of the chosen cluster, and still related to SRP goal and the proposed solution in the cluster, is needed. The requirements analyst will work with the customer to elicit the further requirements needed to fulfil the customer needs. In the *Users Capacity* SRP, assuming the cluster *Users Capacity by Profile* is selected, its parts allow to state the users capacity and its growth (either by amount or by percentage) for different types of user profiles. Nevertheless, it may happen that the customer wants to state other aspects related to the supported users capacity not present in the cluster, like the growth of supported users only when certain conditions are met. In that case new requirements related to the SRP cluster will be defined (alternative scenario B in Figure 5.6).

Requirements Not Related to SRPs

The last situation happens when the SRPs in the catalogue or the SRPs classified in some scope (i.e. classifier of a classification schema) do not cover all the customer needs related to that scope.

Create new requirements not related to SRPs (S.12). This step is necessary in case it does not exist in the catalogue an SRP that solves a need of the customer. As example, let us assume that the customer is interested in specifying the capacity of network traffic that the system under development shall support. As in the SRP catalogue no SRP exists with that goal, new requirements not related to the SRPs have to be extracted (scenario N in Figure 5.6).

CHAPTER 6 Lifecycle of an SRP Catalogue

6.1 Introduction

The SRP catalogue is a lively asset. Once it has been created and experts have checked its quality, the knowledge gained during requirements processes that have used the SRP catalogue should be incorporated into it (Figure 6.1). This update process is important to maintain the SRP catalogue up-to-date; otherwise, when time passes, it will become obsolete and all the effort put into the catalogue construction will not be worth it. As a difference to the use of the SRP catalogue, where requirements analysts are the ones using the catalogue, for both its creation and update RE experts, who are the people in charge of the management of the SRP catalogue, should be the ones taking care of these phases.

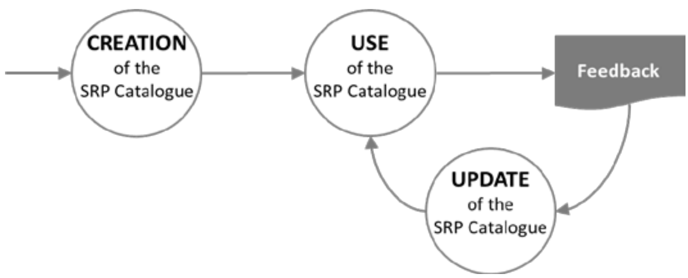


Figure 6.1 – Lifecycle of an SRP catalogue

For the creation of the SRP catalogue, either a bottom-up process or a top-down approach could have been chosen. In bottom-up processes, the requirements are taken as input to build patterns. Top-down approaches, as the one described by Rolland and Prakash [176], go from the general to the particular, starting from concepts to which is desired to build patterns, and afterwards digging-up information about these concepts to build the patterns.

The PABRE method for creating the SRP catalogue follows a bottom-up process. In a nutshell, this method takes as starting point real SRSs that contain requirements stated in natural language, filters the requirements of the selected type (functional, non-functional or non-technical), aligns them according to the concept they refer to, analyses these concepts to study their adequacy as SRPs, and, for each one of the SRP candidates previously identified, builds an SRP. Finally, once all SRPs have been constructed, they are classified according to some classification schema and their relationships are recorded. A complete description of the creation method is found in Section 6.2.

The use of SRPs during requirements elicitation and specification processes has already been explained in Chapter 5. For update purposes, after the requirements analyst has driven these processes for a specific project using the SRP catalogue and the SRS is complete, the knowledge gained in this project must be capitalized in the SRP catalogue. The requirements analyst will collect the information useful for this purpose, both failures and success on SRP applications. For failures, the situations in which the catalogue does not contain all the information needed in the project are recorded. For successes, each application of an SRP is registered. There exist also cases in the middle, like when an SRP is selected but there is no cluster that captures its goal in the appropriate terms for the customer: this is a success from the SRP point of view (because the SRP has been chosen), but a failure from the cluster point of view (since a cluster is still missing). These situations in the middle are also collected with update purposes. Section 6.3 provides a complete explanation of the update process of the SRP catalogue inside the PABRE framework.

6.2 Creation of an SRP Catalogue

The PABRE framework proposes a bottom-up approach for the creation of SRPs, i.e. real requirements are taken as input to build SRPs. Before beginning the process of extracting patterns, it is necessary to decide for which type of requirements SRPs will be created. As an example of types, the classification of requirements proposed in Section 1.1 could be used, which distinguishes between FRs, NFRs and NTRs. As FRs are not domain independent, in case of wanting to construct SRPs for FRs, the analysis should be specific for a concrete software domain.

As an overview, the method to build patterns for a specific type of requirement (Figure 6.2) follows the stages outlined below:

1. Extract requirements of the desired type.
2. Align requirements according to the concept they refer to.

3. Insert in the catalogue of SRP candidates the requirements and perform the analysis of the adequacy of the requirements as SRPs.
4. Formulate SRPs and insert them in the SRP catalogue.
5. Give structure to the SRP catalogue and perform a final quality study.

With this method, an SRP catalogue is built using mostly SRSs, but also literature review and expert judgment when considered necessary. In the following, each one of the stages of the method are defined in detail. Figure 6.3 shows the steps carried out in each one of these stages. For a matter of clarity, the method will be explained jointly with examples based on the experience of building the NT-SRP catalogue presented in Section 7.3.

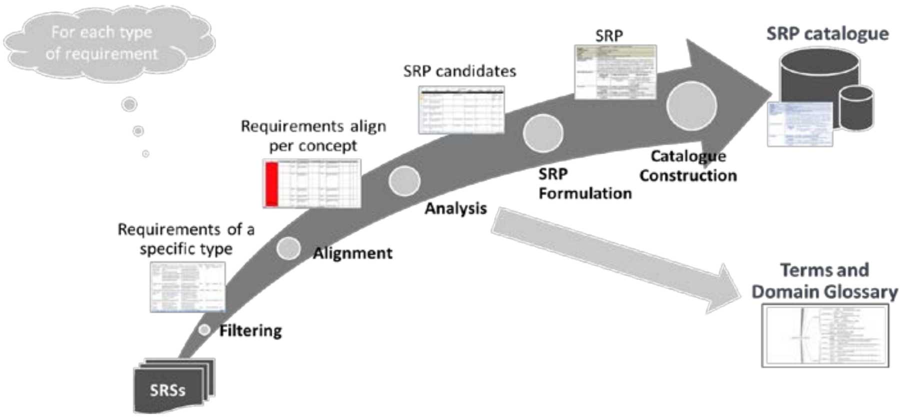


Figure 6.2 – Overview of the SRP creation method

1. Requirements Filtering

Extraction of Requirements of the Selected Type (S.1.1). The requirements that appear in SRSs are of three types (FRs, NFRs, and NTRs) and it is necessary to extract only those ones that are of the desired type, creating a set of target requirements that will evolve until the SRPs are created. This set of target requirements contains requirements as they are found in the SRSs, as well as its location in the document. This allows to have a reference to the origin of the requirement and to make future queries (see Figure 6.4).

In the case of the NT-SRP catalogue, 6 SRSs were used to extract requirements. The SRSs were provided by LIST, as a result of the collaboration established with GESSI. In these SRSs, there were specific sections that were supposed to separately contain NFRs and NTRs. However, during the construction of the NF-SRP catalogue (which was built before the NT-SRP catalogue), some NTRs were discovered in the NFRs sections, and as a consequence 4 NT-SRPs were created while constructing the NF-SRP catalogue: *Crash Response*, *Delivered Documents*, *Document Characteristics* and *Help Desk*. Therefore, the NT-SRP catalogue had already 4 SRPs before the beginning of the process.

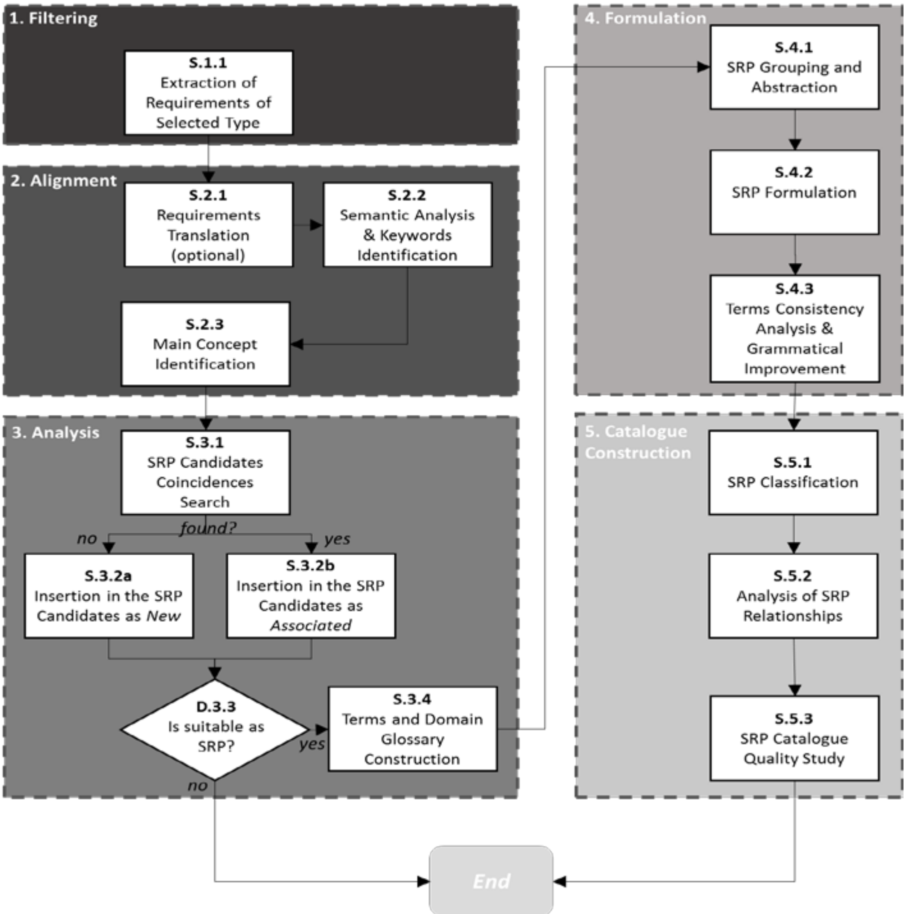


Figure 6.3 – Activities carried out during the SRP creation method

Ref.	1. cdc_RB_CMS_Administration
D.1 Renseignements administratifs	Nous voulons disposer des informations suivantes : - Nom de la société - Adresse de la société - Nom de la personne ayant rédigé l'offre - Numéro de téléphone direct de la personne ayant rédigé l'offre - Numéro de fax de la personne ayant rédigé l'offre - E-mail de la personne ayant rédigé l'offre.

Figure 6.4 – Sample of requirements extracted from SRSs

2. Requirements Alignment

Once the first subset of target requirements is achieved, it is necessary to consolidate and align the requirements according to the concept they refer to. This stage is needed in order to cluster the requirements for pattern identification purposes.

Translation of requirements (S.2.1). The target requirements are expressed in natural language (see Figure 6.4). The first step is, if necessary, to translate these requirements into a known language. With that purpose, different tools such as translators and dictionaries can be used.

The NTRs extracted from the SRSs were written in French. Therefore, it was necessary to do a translation of the requirements to English, trying not to change their meaning and semantics. For that purpose, the translation was reviewed by LIST members, who are experts in both languages.

Semantic analysis and keywords identification (S.2.2). The same requirement may be expressed differently in each of the SRSs or may contain more than one requirement expressed in it. Because of this, it is necessary to study each requirement by means of a semantic analysis. This semantic analysis (currently manual) allows identifying and understanding the concepts presented in each requirement, dividing the requirement into the various restrictions found in it, and assigning keywords for each one of the individual requirements identified.

Table 6.1 – Sample of requirements after Requirements Alignment is finished

Main Concept	Requirement	Keywords	SRSs Id	Reference
Maintenance (maintenance duration)	From the date of expiry of the warranty period, the contractor agrees to provide, at the explicit request of the client, ongoing maintenance services for a minimum period of one year.	expiration, maintenance period, ongoing maintenance, warranty	3	4.5.4
Maintenance (maintenance duration)	The proposed solution must be maintained for at least 1 year from the date of expiration of the warranty period.	expiration, maintenance period, warranty	2	D.4.4
Maintenance (maintenance duration)	The solution should be maintained for 3 years from the expiration of the warranty period.	expiration, maintenance period, warranty	5	D.2.2.3
Steering Committee (frequency)	At least once a month, project stakeholders will meet to evaluate the good progress of the project.	evaluation, progress, project, steering committee	1	D.3.7
Steering Committee (progress report)	At each steering committee, a statement of progress will be prepared and signed by the parties.	progress, report, sign, steering committee	1	D.3.7

For the NT-SRP catalogue, 785 requirements and 54 concepts were identified in this step. Table 6.1 contains some examples of the situations mentioned before. The three first rows of the table referred to the same restriction (the maintenance duration that shall be provided by the supplier), but they were expressed different in the SRSs. In addition, some requirements of the SRSs had to be broken down in individual requirements, as can be seen in the two last rows of Table 6.1. These two restrictions were included in the corresponding SRSs documents in just one complex requirement.

Identification of the main concept (S.2.3). If more than one concept is found in the same requirement, the main concept for such requirement has to be decided.

During the construction of the NT-SRP catalogue, some concepts were easier to identify than others. The first column of Table 6.1 contains the main concept identified for the requirements sample. In the case of the three first rows, it was easy to decide the main concept, as the requirements were clearly stating a restriction over the *Maintenance*. For the fifth row, it was also easy, being *Steering Committee* the main concept. However, some doubts arose in the case of the fourth requirement. One could consider that the main concept could be *Steering Committee* (since the requirements was specifying that project stakeholders shall meet) or *Project Progress Control* (as the requirement was specifying that the goal of the meeting was to evaluate the progress of the project). At the end, it was decided that the main concept was *Steering Committee*, since the progress report was prepared in these kind of meeting.

Table 6.1 contains the information extracted for each requirement in this stage. It is worth remarking that the location of the requirements (i.e. the SRSs id where they were found as well as the reference of the section of the SRS) corresponds to information coming from the previous stage *Requirements Filtering*.

3. Requirements Analysis

For each one of the concepts identified in the previous stage, a study of their adequacy as SRP is performed. To do that, each requirement is inserted in a catalogue of SRP candidates as a candidate requirement. This requires identifying whether the candidate requirement already exists in the catalogue searching for coincidences (i.e. identifying similar requirements) based on the keywords. Afterwards, the former set of candidate requirements is restricted to a subset with all the concepts (and their associated requirements) that may be considered patterns seeds or SRP candidates. During the analysis, a glossary of terms and possible domain values is built. The requirements analysis is done in the following four steps.

Search of coincidences in the candidates' catalogue (S.3.1). This search of coincidences is made in order to group requirements that later can be formalized as an only SRP. The data collected for each group of candidate requirements that are identified is: id (the id of the SRP candidate), number of associated candidates (the number of requirements of the SRP candidate, without taking into account the appearances in different SRSs documents), number of occurrences (the number of requirements of the SRP candidate, counting the appearances in different SRSs documents), and number of SRSs (the number of different SRSs where the requirements of the SRP candidates appear). Before inserting a requirement into the candidates, it is necessary to validate first if there are coincidences with other requirements already inserted as candidate requirements.

In the case of the NT-SRP catalogue, 54 SRP candidates were identified, with a total of 324 candidate requirements.

Coincidences validation and updating or insertion in the candidates' catalogue (S.3.2a, S.3.2b). In case of not finding coincidences, the requirement is added to the catalogue of candidates as a new candidate (Table 6.2). If coincidences are found, two situations may happen: 1) If the same requirement is found as a coincidence, the SRS ids, references, number of SRSs and number of occurrences are updated, taking into account the information of the requirement being processed (Table 6.3); 2) If none of the coincidences corresponds to the requirement being processed, then the requirement is added as an associated candidate of the requirement to which it matches; this association is reflected by updating the counter of associated candidates, writing the identifier of the candidate that has matched and increasing the number of occurrences and the number of SRSs of this candidate (Table 6.4).

Table 6.2 – Example of new SRP candidate

id	#cand	Main Concept	Requirement	Keywords	SRSs Id	Referen-ces	#SRSs	#occur
2	1	Steering Committee (frequency)	At least once a month, project stakeholders will meet to evaluate the good progress of the project.	evaluation, progress, project, steering committee	1	D.3.7	1	1

Table 6.3 – Example of associated SRP candidate that does not imply the addition of a new candidate

Id	#cand	Main Concept	Requirement	Keywords	SRSs Id	Referen-ces	#SRSs	#occur
2	1	Steering Committee (frequency)	At least once a month, project stakeholders will meet to evaluate the good progress of the project.	evaluation, progress, project, steering committee	1 2	D.3.7 D.6.7	2	2

Table 6.4 – Example of associated SRP candidate that implies the addition of a new candidate

Id	#cand	Main Concept	Requirement	Keywords	SRSs Id	Referen-ces	#SRSs	#occur
2	1	Steering Committee (frequency)	At least once a month, project stakeholders will meet to evaluate the good progress of the project.	evaluation, progress, project, steering committee	1 2	D.3.7 D.6.7	2	3
2	2	Steering Committee (progress report)	At each steering committee, a statement of progress will be prepared and signed by the parties.	progress, report, sign, steering committee	1	D.3.7		

Table 6.5 – Sample of SRP candidates after Requirements Analysis is finished

id	#cand	Main Concept	Requirement	Keywords	SRSs Id	Referen-ces	#SRSs	#occur
1	1	Maintenance (maintenance duration)	From the date of expiry of the warranty period, the contractor agrees to provide, at the explicit request of the client, ongoing maintenance services for a minimum period of one year.	expiration, maintenance period, ongoing maintenance, warranty	3	4.5.4	6	6
1	2	Maintenance (maintenance duration)	The proposed solution must be maintained for at least 1 year from the date of expiration of the warranty period.	expiration, maintenance period, warranty	2 4 6	D.4.4 D.4.4 D.4.7		
1	3	Maintenance (maintenance duration)	The solution should be maintained for 3 years from the expiration of the warranty period.	expiration, maintenance period, warranty	5 1	D.2.2.3 D 2.2.3		
2	1	Steering Committee (frequency)	At least once a month, project stakeholders will meet to evaluate the good progress of the project.	evaluation, progress, project, steering committee	1 2 4	D.3.7 D.6.7 D.5.2	3	6
2	2	Steering Committee (progress report)	At each steering committee, a statement of progress will be prepared and signed by the parties.	progress, report, sign, steering committee	1 2 4	D.3.7 D.6.7 D.5.2		

Table 6.5 contains the SRP candidates extracted during this stage for the sample of NTRs of Table 6.1. The concept *Maintenance* was found six times in six different SRSs.

Notice that in this concept, there were three candidates, but the three of them corresponded to the same type of restriction (i.e. the duration of the maintenance). The concept *Steering Committee* was found six times in three different SRSs. In this concept, however, the two candidates represented two different types of restrictions (i.e. the frequency in which the steering committee meetings shall be done, and the necessity of writing in these meetings a progress report).

Study of the suitability as SRP (D.3.3). Afterwards, it is analysed (based on the number of occurrences and the expert assessment) if the candidate is suitable for being an SRP or not. The main criterion of course is repetition, because that identifies high probability of reuse: those requirements that appear in most or all of the analysed SRSs are clear candidates. For that, the counter of occurrences and the number of different SRSs in which the candidate appear are used. However, this is not the only criterion. A requirement appearing in a few, even just one, SRS may also be considered adequate as SRP. In this step, expert assessment is the cornerstone, since experts are the only ones that may say, for instance, that a requirement appearing in just one SRS could in fact have appeared in all of them. If the candidate being analysed is not a candidate suitable for being an SRP, the process is finished for this candidate, and the process continues with another candidate.

Table 6.6 – Example of SRP candidate appearing in one SRS

Id	#cand	Main Concept	Requirement	Keywords	SRSs Id	Referen-ces	#SRSs	#occur
3	1	Audit (right)	The *customer* reserves the right to conduct audits of the provider and its production during the project.	quality, audits, provider, project production	4	D.6.7	1	2
3	2	Audit (focus)	These audits will focus on the specific development (product code, development methodology, and documentation), the treatment of reported anomalies and the quality procedures.	quality, audits, development, anomalies	4	D.6.7		

During the construction of the NT-SRP catalogue, 47 SRP candidates were identified as suitable for being converted into SRP, with a total of 298 candidate requirements. From Table 6.5, both *Maintenance* and *Steering Committee* were two of them. In addition, SRP candidates appearing in just one SRS, as the one in Table 6.6, were considered suitable for being SRP, as even if they were appearing in just one SRS, experts considered that other projects could benefit of these requirements, and therefore it made sense to have them encapsulated in an SRP.

Building the terms and domains glossary (S.3.4). During the analysis, it is important to build a glossary of terms and possible domain values to be used in the future. The idea is to take note of those possible terms that are recurrent in the requirements, in order to homogenize them when formulating the SRPs.

For the NT-SRP catalogue, 249 items were part of the terms and domain glossary. Some of these items were: *Customer* (instead of client, purchaser, etc.), *Supplier* (instead of manufacturer, producer, provider, etc.), and *System* (instead of approach, proposal, solution, etc.).

4. SRP Formulation

In this stage, the SRP candidates are converted into SRPs following the next three steps.

Abstraction and grouping of SRPs (S.4.1). The different SRP candidates are converted into SRPs mainly by means of abstraction. One way of giving abstraction to the text of the SRP candidates is substituting specific aspects related to one project by parameters related to certain domains. In addition, not every candidate is necessarily converted into a different SRP, since some of them may be considered close enough as to be integrated in the same pattern.

Table 6.7 – Sample of SRP abstractions during SRP Formulation (I)

id SRP	id Req. Abstraction	Main Concept	Requirement Abstraction	Keywords	SRSs Id	References
1	1	Maintenance (maintenance duration)	The supplier shall offer *maintenanceType* maintenance for the system for a minimal duration of *amountOfTime timeUnit*.	expiration, maintenance period, warranty	1 2 3 4 5 6	D.2.2.3 D.4.4 4.5.4 D.4.4 D 2.2.3 D.4.7
2	1	Steering Committee (frequency)	The steering committee shall meet during the system implementation project every *amountOfTime timeUnit*.	evaluation, progress, project, steering committee	1 2 4	D.3.7 D.6.7 D.5.2
2	2	Steering Committee (progress report)	The supplier and customer shall sign a project progress report after each steering committee meeting to validate it.	progress, report, sign, steering committee	1 2 4	D.3.7 D.6.7 D.5.2
3	1	Audit (right)	If the customer considers it necessary during the system implementation project, it shall be allowed to audit the process or of the *projectDeliverables*.	quality, audits, provider, project production	4	D.6.7
3	2	Audit (focus)	The customer shall focus the audit on the *qualityAspects* of the *projectDeliverables*.	quality, audits, development, anomalies	4	D.6.7

Table 6.8 – Sample of SRP abstractions during SRP Formulation (2)

id SRP	id Req. Abstraction	Main Concept	Requirement Abstraction	Keywords	SRSs Id	References
1	2	Corrective Maintenance (maintenance tasks)	The corrective maintenance shall include *maintenanceTasks* for the implemented system.	expiration, maintenance period, warranty	2 6	D.5.2 D.3.4

In the case of the NT-SRP catalogue, from the 47 SRP candidates identified as suitable for being converted into SRPs, 41 SRPs were built. Table 6.7 contains the abstraction of the SRP candidates of Table 6.5 and Table 6.6. For instance, the first SRP in the table corresponds to the abstraction of the three first SRP candidates of Table 6.5. Abstraction was added to the candidates in order to allow the statement of different periods of maintenance for the different types of maintenances (e.g. corrective or functional). During this process, some SRP candidates were merged together in the same SRP. This is the case of the first row of Table 6.7 and Table 6.8. As their main concepts were clearly related (both were dealing with maintenance issues) and corrective maintenance was one of the possible maintenance types of the requirements abstraction corresponding to the first row of Table 6.7, it was decided to merge both SRP candidates in the same SRP, giving place to the *Maintenance Types* SRP. In addition, the *Maintenance Types* SRP also was built using other candidates related to the different maintenance types that a supplier could provide.

Formulation of the SRPs (S.4.2). Afterwards, the final structure of the SRPs, their clusters, parts and parameters, emerges. In this process, again with expert assessment, the final structure of every SRP may be slightly different than the corresponding requirements in the SRSs, since experts may consider that for future projects these differences could be useful.

At this point, the 4 NT-SRPs extracted while building the NF-SRP catalogue were added to the NT-SRP catalogue. After the formulation, the NT-SRP catalogue was composed of 45 SRPs, with a total of 56 clusters, 211 parts and 304 parameters. Next, a couple of examples about the formulation of the NT-SRPs are presented.

The requirement abstractions related to *Audit* (the last two rows of Table 6.7) were included in the catalogue as just one SRP, together with other requirement abstractions that specifically addressed audits done following a certain quality standard. Therefore, the SRP was structured in two alternative clusters: the *General Audit* cluster and the *Standard-Based Audit* cluster. In the case of the general cluster, the first requirement abstraction was selected as the fixed part of the cluster and the second as the extended part, since the second requirement did not have sense in a project if the first requirement did not appear too.

During this step, it was noticed that there was an SRP candidate restricting the *Documentation* of the project. In this case, it was observed that two of the first 4 NT-SRPs already were dealing with the documentation of the project (the *Delivered Documents* SRP and the *Document Characteristics* SRP). Thus, it was decided that some of the SRP candidates were redundant regarding to the existing NT-SRPs, and other had to be added to them as new extended parts.

Analysis of the consistency of terms and improvement of the grammar (S.4.3). For each SRP, a term consistency analysis (using the glossary constructed in the previous stage) and grammatical improvement is applied. For instance, for the templates of the parts, syntactical conventions are enforced, making all the requirements to have the same structure.

In order to give consistency to the NT-SRP catalogue, the glossary of terms and domains built in the previous stage was used. It is important to highlight that this glossary could evolve while using it. For instance, when a new domain or domain value was needed, it was added to the glossary (that avoided having redundant terms or domains). In the example, the domains *amountOfTime* and *timeUnit*, as well as the terms *supplier* and *customer*, used in some of the requirement abstractions of Table 6.7 and Table 6.8, already existed in the glossary. Some grammatical rules were applied too (as seen in the requirement abstractions of Table 6.7 and Table 6.8): the abstractions were written in an active voice, in third-person and using the modal verb *shall*.

5. SRP Catalogue Construction

Finally, the SRPs evolve from individual artefacts into an articulated structure of knowledge stored in the catalogue. The following three things need to be done.

SRP classification (S.5.1). The SRPs are classified according to classification schemas. The classification schemas used can be created on this step, or can already exist.

For the NT-SRP catalogue, three different classifications were used: one based on the ISO/IEC 25010 standard system and software quality model catalogue [123], one based on the ISO/IEC 9126 standard software quality model catalogue [128] (the ancestor of the ISO/IEC 25010), and one corresponding to the sections used in the SRSs from where the SRPs were extracted, which were based on the Volere approach presented by Robertson and Robertson [145]. As the characteristics and subcharacteristics of the ISO/IEC 25010 the ISO/IEC 9126 standards do not address non-technical aspects of software, an adaptation of the NT-ISO/IEC 9126 catalogue proposed by Carvallo et al. [8] was used as a base and adapted in the ISO/IEC 25010-based and ISO/IEC 9126-based classification schemas. More information about this adaptation, as well the ISO/IEC 25010-based classification schema, can be found in Section 7.3.

During the classification of the NT-SRPs, both the *Steering Committee* SRP and *Audit* SRP were classified as patterns related to the project as a business (i.e. *Project* root classifier), and more specifically as patterns related to the relationships established among the customer and the supplier (i.e. *Supplier Relationship* classifier). In addition, some doubts arose when classifying some patterns. For instances, while classifying the *Maintenance Types* SRP, there was a doubt since there existed in the ISO/IEC 25010 and ISO/IEC 9126-1 a subcharacteristic named *Maintenance*, and at first it was thought that the SRP had to be placed there. However, after checking the meaning of the subcharacteristic, it was clear that it had a different meaning than the one of the SRP, since the subcharacteristic was supposed to group requirements related to how the system should be for being more maintainable. Thus, the *Maintenance Types* SRP was classified inside the root classifier *Supplier* and the classifier *Support*, since the SRP was related to a support service offered by the supplier.

Analysis of relationships among SRPs (S.5.2). The relationships among SRPs are established and recorded. For establishing the relationships, an analysis is done over the SRPs looking at their names, goals, keywords, clusters' names, extended parts' names, templates and parameters. In this analysis, what is being looking for is common or similar concepts, or contradictory ones, that make arise the finding of a relationship.

More than 125 relationships were found in the NT-SRP catalogue, of different granularity (e.g. among SRP and SRP, and among parameter and parameter) and of different types (e.g. *relates-to*, *requires* and *supports*). For finding the relationships, a graph was created, using the yEd tool[20], with all the SRPs and their respective elements (clusters,

[20] http://www.yworks.com/products/yed

parts, parameters and domains). Specifically, a different node was created for each one of the elements. Therefore, as all the information about the SRPs was in the graph, and the tool provided functionalities that ease to see only some specific elements (either searching by their name or by their type), and some SRPs shared parameters and domains, it was easy to see how the SRPs related to each other. One of the relationships identified was the one of type *relates-to* among *Maintenance Types* SRP and *Maintenance Procedures* SRP, since both of them were related to maintenance in different ways: the first one set the types of maintenance that the supplier shall provide and some other restrictions related to them (e.g. the maintenance duration), and the second one expressed the need of assessing the supplier maintenance teams and maintenance experience..

Study of the quality of the SRP catalogue (S.5.3). The quality of SRSs that will be obtained in applying SRPs has to be analysed. Thus, the writing style, consistency and relationships among SRPs are double-checked in order to avoid poor requirements generated from the patterns. For that activity, usually two iterations are needed:

1. The first iteration has the goal of making uniform the granularity of SRPs. The main task in this iteration corresponds to merge SRPs that have similar goals (i.e. solve the same problem) or that restrict the same functionality. Conversely, SRPs that try to give solution to more than one problem or that restrict different functionalities have to be identified and split into several SRPs.
2. The second iteration focuses on aligning the contents of the SRPs along two directions. First, ensuring the consistent use of the glossary of terms and domains built during the *Requirements Analysis* stage. Second, checking some predefined grammatical rules on the template and taking corrective actions in case those deviations are found.

The validation of the NT-SRP catalogue was done by LIST's requirement engineers, which had wide experience in requirements elicitation and specification. First of all, there was a global observation that the experts did, corresponding to the generic approach of the obtained NT-SRPs, and mainly of the ones placed under the *Supplier* classifier. The fact was that most of the patterns were more asking for information about the supplier than restricting how the supplier had to be or had to behave. To have more mature SRPs, experts proposed to formulate the patterns in a more prescriptive way. For instance, the *Supplier*

Workforce SRP had first as goal *"Having information about the supplier workforce"* and the fixed part of its only cluster was *"The supplier shall provide workforce information about the company"*. Taking the experts' advice, the goal of the SRP changed to *"Assessing the workforce of the supplier"* and it was decided to add a new cluster in the SRP establishing a restriction of the supplier workforce, which had as fixed part *"The supplier shall fulfil some workforce requirements"*. Both clusters had extended parts to establish different aspects of the workforce information that had to be obtained or restricted, such as the workforce distribution.

In addition, during the validation, some pairs of NT-SRPs were joined, or some of them were removed because they were redundant with respect to other patterns. Specifically, this happened in the case of the *Installation* SRP, which was subsumed by the *System Implementation Scheduling* SRP, since this last pattern already established the planning of the different activities carried out during the system lifecycle, including the installation. If the *Installation* SRP would have not been removed, that could have driven to non-consistent SRSs.

The changes in the vocabulary and the abstraction of the NT-SRPs from data of specific contexts of application continued during the validation. In the case of the *Audits* SRP, the experts decided to change in the term *"audit"* by *"assess of the quality"*, as the last term identified better the purpose of the pattern, giving place to the *Quality Assessment* SRP.

After the validation, the current version of the NT-SRP catalogue (which can be found in Section 7.3) was obtained. It was composed of 38 SRPs, with a total of 48 clusters, 194 parts and 285 parameters, and 129 relationships were identified on it.

6.3 Update of an SRP Catalogue

The evolution of an SRP catalogue allows keeping the catalogue up-to-date, by capitalizing the knowledge gained with the different projects that have used the SRP catalogue for the elicitation and specification of requirements.

To achieve that, the requirements analyst will collect the information useful for this purpose, both successes and failures on SRP application, and also cases in the middle. Then, this information will be used by the RE expert to take the decision on whether or not to take the actions to update the SRP catalogue. In some cases, the RE expert may need to check with the requirements analyst before taking the decision, as the requirements analyst is the person using the SRP catalogue on the field. Figure 6.5 shows both the information generated during the requirements elicitation and specification processes used for the update, and also the actions taken by the RE expert to update the catalogue.

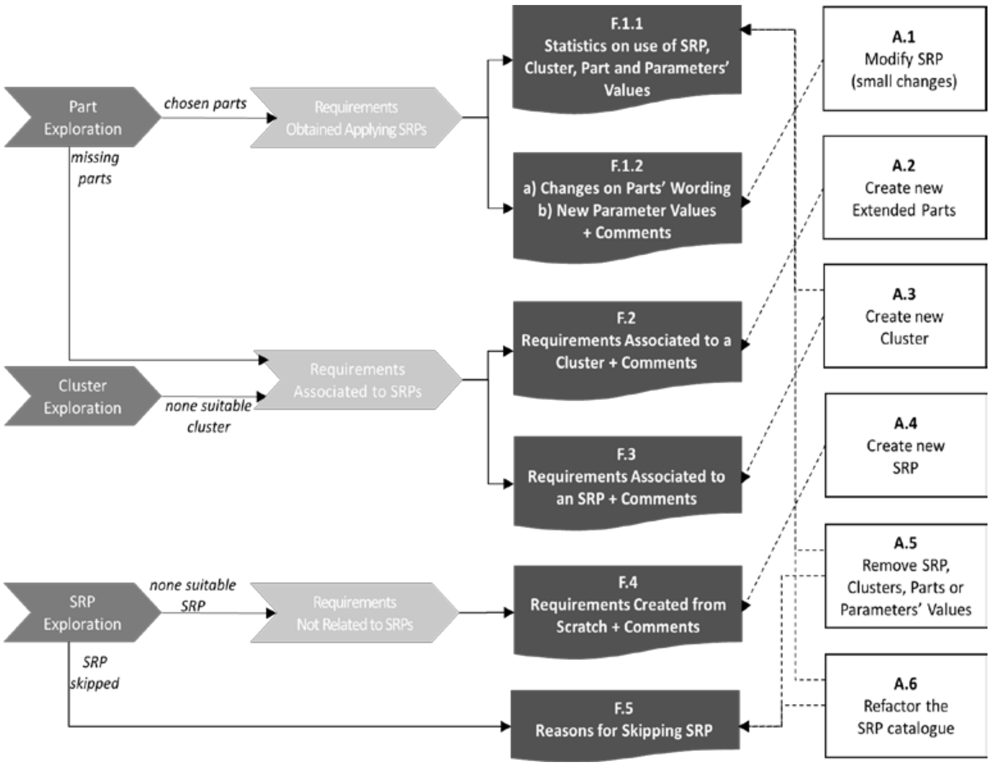


Figure 6.5 – Feedback information and actions taken for the update of the SRP catalogue

The information gathered by the requirements analyst during the requirements elicitation and specification process is:

- For successes, each application of an SRP is recorded (feedback information F.1.1). This information includes the statistics on the use of the chosen SRP, clusters, parts and values of the parameters. If small changes have been done in the wording of the parts, or if new parameters' values not existing on the catalogue have been needed, these changes or needs are recorded in feedback F.1.2. This feedback also includes the comments that requirements analysts can give about why a change in the wording of a part or new parameters have been needed. These comments could be directly related to the quality of the SRP, but also to other aspects.

- For the cases in the middle, when an asset in an SRP is still missing, the feedback will contain the new requirements that have been created because of this lack in the SRP as well as the comments that requirements analysts can give about why new requirements have been needed (in terms of SRP quality or other aspects). There are two situations that are considered cases in the middle:
  - When an SRP and one of its clusters are chosen, but some extended part is missing: this is a success from the SRP and cluster perspective (since both have been chosen), but a failure for the extension (since an extended part is still missing) (F.2).
  - When an SRP is selected but there is no cluster that captures its goal in the appropriate terms for the customer: this is a success from the SRP point of view (i.e. the SRP has been chosen), but a failure from the cluster point of view (since a cluster is still missing) (F.3).
- For failures, the situations in which the catalogue does not contain an SRP that captures a need of the project at hand are recorded; specifically, the new requirements created to fulfil this need are stored as well as the requirements analysts' comments about why new requirements have been needed (F.4).

Finally, there is also another situation considered neither success nor failure: when SRPs are skipped. In this case, the requirements analyst collects the reason for skipping the SRPs (F.5), which could be related to either a quality aspect of the SRP or to a specific aspect of the project.

The different actions that the RE expert may take to enlarge the SRP catalogue, and the situations in which these actions may be taken, are:

- Promote set of requirements into SRPs (A.4), when requirements have been written from scratch, using the feedback information F.4 to create new SRPs. However, before doing that, the RE expert has to analyse if there has been an error in defining these requirements as new, or if in fact the requirements analyst is right and there is no SRP in the catalogue that corresponds to the goal of the new requirements.
- Create new clusters (A.3), when requirements are related to an SRP goal but they are not related to any existing cluster, using the feedback information F.3 and F.1.1. If the number of direct applications of an SRP is low regarding to its

associations, the RE expert has to check the associated requirements in order to find out if there is some problem with the definition of the SRP.

- Create new extended parts (A.2), when requirements are related to a cluster but the details needed are not contained in any existing extension. For creating these new extensions, the information F.2 is revised to check if there are various projects needing a similar extension for a specific cluster.
- Modify parts (A.1), when parts have been applied but small modifications have been needed, either in the wording of the parts or new values for parameters have arose in the project. The information F.1.2 is used for that purpose. The RE expert checks if various projects have done a similar modification in a part or have added the same value to a parameter, and then the modification is applied.

However, not just enlargement is possible. Some other operations can also be applied over the catalogue after updating its statistics of use with the data of a project. By using the information stored in F.1.1 and F.5, the RE expert identifies the SRPs, clusters, extended parts and parameters which are the most and least used. This information is used to remove unused SRPs, clusters, extended parts or parameters' values that do not seem to be relevant anymore (A.5), or even refactoring of the catalogue by splitting or joining some SRPs (A.6).

In all the situations explained, even if the criteria for doing the modifications over the SRP catalogue are met, is the RE expert who has the last word on taking the action or not, so if s/he believes the modification is not of value to the SRP catalogue, the modifications will not be done.

CHAPTER 7 Catalogue of SRPs

7.1 Introduction

The current form of the SRP catalogue embraces 28 NF-SRPs, 38 NT-SRPs and 44 F-SRPs. As functional requirements are not domain independent, the F-SRPs are focused on a concrete domain, specifically the CMS domain.

The catalogue was constructed using the SRP construction process presented in Section 6.2, and using 6 SRSs as starting point of the process. These SRSs were provided by LIST as a result of the collaboration established with GESSI.

In these 6 SRSs, the main sections were supposed to separately contain FRs, NFRs and NTRs. However, when building the SRP catalogue for NFRs, it was discovered that this separation was not strict, since some NTRs were discovered in the NFRs section. As a result, while constructing the NF-SRP catalogue, 4 NT-SRPs were identified, which became the initial set of the NT-SRP catalogue. These SRPs are: *Crash Response*, *Delivered Documents*, *Document Characteristics* and *Help Desk*.

The requirements in the SRSs were written in French. However, the biggest core of knowledge on RE is available in English. In addition, for dissemination purposes, the goal was to produce the SRP catalogue in English. Therefore, during the alignment process, the requirements were translated into English. The translation was supervised by the LIST team since French is their native language but they are also fluent in English.

Currently there are three classification schemas for organizing the catalogue: the first one is based on the ISO/IEC 25010 standard system and software quality model catalogue [132], the second one is based on the ISO/IEC 9126 standard software quality model catalogue [137] (the ancestor of the ISO/IEC 25010), and the third one is the classification schema used for the requirements in the analysed SRS documents, which is based on the Volere approach presented by Robertson and Robertson [155].

Section 7.2, Section 7.3 and Section 7.4 present the NF-SRP, NT-SRP and F-SRP parts of the catalogue, respectively. The catalogue is presented using the ISO/IEC 25010-based schema. A complete version of the catalogue containing all the SRPs and classification schemas can be found in http://www.upc.edu/gessi/PABRE/SRPCatalogue.pdf.

7.2 Non-Functional SRP Catalogue

The NF-SRP catalogue obtained from the 6 analysed SRS documents is composed of 28 SRPs. The coverage of the catalogue in the three available classification schemas is: it covers 15 out of 46 subcharacteristics of the second level of the ISO/IEC 25010-based classification schema, 13 out of the 42 of the ISO/IEC 9126-1-based classification schema, and 20 out of the 49 of the Volere-based classification schema. The NF-SRP catalogue organized using the classification schema based on the ISO/IEC 25010 is shown in Table 7.1.

Table 7.1 – NF-SRP catalogue classified using the ISO/IEC 25010-based classification schema

ISO/IEC 25010 Classifiers		SRP	Entity Types	#Clusters / #Extended Parts	#Parameters
Functional Suitability	Functional Completeness	-	-	-	-
	Functional Correctness	Data Precision	Data Type, Subsystem	2 Cl / (2 EP, 0 EP)	(1, 2, 2) (4)
	Functional Appropriateness	-	-	-	-
Performance Efficiency	Time Behaviour	Interface Load Time	Action, Subsystem	2Cl / (0 EP, 0 EP)	(3) (4)
	Resource Utilization	-	-	-	-
	Capacity	Concurrent Users Capacity	Subsystem	1Cl / (0 EP)	(2)
		Data Capacity	Subsystem	1Cl / (1 EP)	(3, 3)
		Users Capacity	User profile, Subsystem	2Cl / (2 EP, 2 EP)	(2, 2, 2) (3, 3, 3)
Compa-tibility	Co-existence	-	-	-	-
	Interoperability	Data Exchange	Subsystem	1Cl / (5 EP)	(1, 2, 3, 2, 3, 4)
		Interoperability with External Systems	Subsystem	1Cl / (4 EP)	(1, 2, 2, 3, 2)
Usability	Appropriateness Recognisability	-	-	-	-
	Learnability	Interface Learnability	Subsystem	1Cl / (1 EP)	(2, 5)
		Online Help (r)	Subsystem	1Cl / (4 EP)	(1, 2, 2, 2, 1)
	Operability	*Failure Alerts (r)*	Subsystem	2Cl / (2 EP, 1 EP)	(1, 2, 2) (1, 3)
		Installation Procedures (r)	Subsystem	2Cl / (2 EP, 1 EP)	(1, 2, 1) (1, 1)
		Online Help (r)	Subsystem	1Cl / (4 EP)	(1, 2, 2, 2, 1)
		Recovery Procedures	Subsystem	1Cl / (2 EP)	(1, 2, 2)
		Update Procedures (r)	Subsystem	1Cl / (2 EP)	(1, 1, 2)
	User Error Protection	-	-	-	-

	User Interface Aesthetics	-	-	-	-
	Accessibility	Interface Language	Subsystem	1CI / (0 EP)	(2)
		Interface Type	Subsystem	1CI / (1 EP)	(2, 3)
Reliability	Maturity	*Failure Alerts (r)*	Subsystem	2CI / (2 EP, 1 EP)	(1, 2, 2) (1, 3)
	Availability	Availability	Subsystem	1CI / (0 EP)	(2)
		Downtime	Subsystem	1CI / (1 EP)	(4, 3)
		Uptime	Subsystem	1CI / (1 EP)	(3, 3)
	Fault Tolerance	Alternative Data Storage	Subsystem	1CI / (0 EP)	(2)
	Recoverability	Backups	Subsystem	1CI / (4 EP)	(1, 2, 3, 2, 2)
		Logs	Subsystem	1CI / (3 EP)	(1, 2, 2, 2)
Security	Confidentiality	Authorization	Subsystem	1CI / (9 EP)	(1, 1, 4, 1, 2, 2, 2, 1, 1, 2)
		Automatic Logoff	Subsystem	1CI / (2 EP)	(1, 1, 3)
	Integrity	Authentication	Subsystem	1CI / (3 EP)	(1, 1, 2, 2)
		Data Transmission Protection	Subsystem	1CI / (4 EP)	(1, 2, 2, 2, 2)
		Stored Data Protection	Subsystem	1CI / (1 EP)	(2, 2)
	Non-repudiated	-	-	-	-
	Accountability	-	-	-	-
	Authenticity	-	-	-	-
Maintai-nability	Modularity	-	-	-	-
	Reusability	-	-	-	-
	Analysability	-	-	-	-
	Modifiability	-	-	-	-
	Testability	-	-	-	-
Portability	Adaptability	Development Language	Subsystem	2CI / (2 EP, 2 EP)	(3, 4, 4) (3, 4, 4)
	Installability	*Installation Procedures (r)*	Subsystem	2CI / (2 EP, 1 EP)	(1, 2, 1) (1, 1)
		Platform	Subsystem, Technology type	3CI / (0 EP, 0 EP, 0 EP)	(3) (3) (2)
		Update Procedures (r)	Subsystem	1CI / (2 EP)	(1, 1, 2)
	Replaceability	-	-	-	-
Business	Licensing Schema	-	-	-	-
	Ownership	-	-	-	-
	Guarantees	-	-	-	-
	Costs	-	-	-	-
Product	History	-	-	-	-
	Deliverables	-	-	-	-
	Parameterization and Customization	-	-	-	-
Project	Business Scheduling	-	-	-	-
	Supplier Relationships	-	-	-	-
Supplier	Organizational Structure	-	-	-	-
	Positioning and Strength	-	-	-	-
	Reputation	-	-	-	-
	Services Offered	-	-	-	-
	Support	-	-	-	-
Functio-nal Reqs.	Content Management Systems	-	-	-	-

As it can be seen in Table 7.1, four SRPs are situated below two classifiers: *Failure Alerts*, *Installation Procedures*, *Online Help* and *Update Procedures*, because, for instance, requirements corresponding to the *Failure Alerts* SRP can be included in the *Operability* section of one SRS document, and in the *Maturity* section of another.

The table also includes, for each NF-SRP, the *Entity Types*, the number of *Clusters* and *Extended Parts* of each SRP, as well as the number of *Parameters* for each part. The *Concept* for each SRP is not stated, since it is the same than the name of the SRP, neither the number of *Fixed Parts*, since it is the same than the number of clusters. Note that the SRPs have among 1 and 3 clusters, each cluster has among 0 and 9 extended parts, and each part has at least 1 and at most 5 parameters.

The number of relationships found on the NF-SRP catalogue is shown in Table 7.2. The most frequent relationships are those ones among *SRP* and *SRP*, and among *Parameter* and *Parameter*, highlighting in the first case the relationships of type *relates-to* (34 relationships) and *supports* (11 relationships), and in the second case remarking the relationships of type *supports* (48 relationships).

Table 7.2 – Number of relationships of the NF-SRP catalogue classified per type and elements

Relationship Elements	Conflicts	Disjoint	Relates-to	Requires	Rule	Subset	Supports
SRP - SRP	1	-	34	4	-	-	11
Cluster- SRP	-	-	-	1	-	-	-
Part - SRP	-	-	-	1	-	-	-
Domain - SRP	-	-	-	2	-	-	-
Part - Part	-	-	4	-	-	-	-
Param - Param	-	2	-	-	1	2	48

As illustration, an NF-SRP is shortly presented, the *Failure Alerts* NF-SRP (Table 7.3). This SRP produces requirements related to the goal of having a system that *Alerts users about system failures* (what types of failures shall be monitored, what types of alerts shall be launched, etc.). Although the SRP describes a functionality of the system, it is still considered an NF-SRP. As defined in Subsection 1.1.1 of this thesis, FRs are the ones that define functionalities that the system shall offer. The *Failure Alerts* SRP, however, does not provide a functionality offered to the user but a functionality that is considered, according to the ISO/IEC 25010 standard [132], a quality property inherent to the system related to both its operability[21] and maturity[22], and, as

[21] Operability. *Degree to which a product or system has attributes that make it easy to operate and control* [132].
[22] Maturity. *Degree to which a system meets needs for reliability under normal operation* [132].

a quality property, it is considered to be non-functional. The two clusters of the SRP differ on whether it is necessary to have specific types of alerts depending on the types of failures or not. For the *Homogeneous Failure Alerts* cluster, the fixed part states for which subsystems failures shall be alerted, and the extended parts what type of alerts shall be launched and what type failures shall be alerted for a specific subsystem. Its behaviour explains that the three parts can be applied at most once for each subsystem, since it is better, for instance, to state for the same subsystem just one requirement about what type of failures shall be alerted. In the *Heterogeneous Failure Alerts* cluster, the fixed part states for what subsystems different types of alerts shall be triggered depending on the type of failure, and the extended part establishes what type of alerts shall be triggered for a specific failure in a specific subsystem. The behaviour states that the fixed part can be applied at most once for each subsystem, and the extended part can be applied at most once for a specific subsystem and type of failure, since it is possible to state for the same subsystem a requirements that says that an SMS shall be triggered when a database crash occurs, and another one that states that an e-mail shall be sent when disks are close to their capacity. The domains of the parameters of both clusters correspond to the set of possible subsystems, alert types and failure types. The *Failure Alerts* SRP has two relationships stated:

- The *Failures Alerts* SRP supports the *Platform* SRP, i.e. the technologies defined by both SRPs for a specific subsystem should be consistent and applicable together.
- The failures defined in the *Failures Alerts* SRP must support the failures defined by the *Logs* SRP and *Recovery Procedures* SRP for a specific subsystem, meaning that the failures in the three SRPs could be the same or that at least they should be consistent in their naming.

Table 7.3 – Failure Alerts NF-SRP

SRP *Failure Alerts*	**Goal**	Alerting users about system failures.
	Keywords	System failure, Failure, Alert, Technical support, Crash response
	Description	This SRP expresses the need of having the system functionality to inform users about system failures at the moment the failure occurs.
	Question	Does the customer require different types of alerts depending on the types of failures?
	Authors	GESSI-LIST
	Sources	Specialized literature, SRSs from LIST
	Date	13 January 2016
	Version	v.3
	Entities	Subsystem
	Relationships	- SRP/SRP support relationship. The *failures* and *alerts* defined by *Failure Alerts* SRP must support the technologies defined by *Platform* SRP - Parameter/Parameter support relationship. *failures* parameter could be the same in and/or should be consistent with the applications of the following SRPs: *Logs*, *Failures Alerts*, *Recovery Procedures*

SRP Cluster *Homogeneous Failure Alerts*	**Description**	This cluster does not establish any relationship among the type of alert and the type of system failure. It has extensions to determine the type of system failures and alerts.		
	Behaviour	- *Fixed Part*, *Alert Types*, *Failure Types*: can be applied at most once each for a same subsystem.		
	Fixed Part	**Template**	*The %subsystem% shall trigger an alert in case of system failure.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
	Extended Part *Alert Types*	**Template**	*The %subsystem% shall trigger %alerts% alerts in case of system failure.*	
		Parameter name	**Domain name**	**Domain type**
		alerts	*AlertTypes*	*Set (AlertType=[SMS, Mail, Bip, ...])*
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
	Extended Part *Failure Types*	**Template**	*The %subsystem% shall trigger alerts in case of %failures% failures.*	
		Parameter name	**Domain name**	**Domain type**
		failures	*FailureTypes*	*Set (FailureType=[database crash, network crash, ...])*
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
SRP Cluster *Heterogeneous Failure Alerts*	**Description**	This cluster establishes a dependency among the type of alert and the type of system failure that occurs. The extension establishes which alerts are issued by which system failures.		
	Behaviour	- *Fixed Part*: can be applied at most once for a same subsystem. - *Alerts for Failure Type*: can (usually will) be applied more than once for a same subsystem, provided it is applied for different failures.		
	Fixed Part	**Template**	*The %subsystem% shall trigger different types of alerts depending on the type of system failure.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
	Extended Part *Alerts for Failure Type*	**Template**	*The %subsystem% shall trigger %alerts% alerts in case of %failures% system failures.*	
		Parameter name	**Domain name**	**Domain type**
		alerts	*AlertTypes*	*Set (AlertType=[SMS, Mail, Bip, ...])*
		failures	*FailureTypes*	*Set (FailureType=[database crash, network crash, ...])*
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*

7.3 Non-Technical SRP Catalogue

The NT-SRP catalogue obtained from the 6 analysed SRS documents is composed of 38 SRPs. Table 7.4 contains the NT-SRP catalogue organized using the classification schema based on the ISO/IEC 25010 standard.

As the characteristics and subcharacteristics of the ISO/IEC 25010 standard system and software quality model catalogue and the ISO/IEC 9126 standard software quality model catalogue do not address non-technical aspects of software, an adaptation of the NT-ISO/IEC 9126 catalogue proposed by Carvallo et al. [8] has been used as a base and adapted to the ISO/IEC 25010-based and ISO/IEC 9126-based classification schemas.

The NT-ISO/IEC 9126 extension of Carvallo et al. adds 3 characteristics (*Supplier*, *Business* and *Product*) and 15 subcharacteristics to the standard. Before classifying the NT-SRPs according to this proposal, some changes had to be done to take into account some differences on the use of the catalogue.

On the one hand, NT-ISO/IEC 9126 catalogue was created to include the criteria to assess the quality of a final software system, whereas the NT-SRPs state requisites for the procurement of a system (probably by gluing or adapting several systems). This is the reason why it was needed to add a new characteristic to group the SRPs about the implementation project: the *Project* characteristic, decomposed into two subcharacteristics: *Business Scheduling* and *Supplier Relationships*.

On the other hand, some related subcharacteristics were merged into just one. Specifically, they were those related to the cost of the business. The original subcharacteristics were too static: *Licensing Costs*, *Platform Costs*, *Implement Costs* and *Network Costs*, but the new subcharacteristic integrates all these costs in a cost breakdown structure allowing the flexibility to add new ones.

The coverage of the catalogue is quite similar in the three available classification schemas: it covers 17 out of the 46 subcharacteristics of the second level of the ISO/IEC 25010-based classification schema, 16 out of the 42 of the ISO/IEC 9126-1-based classification schema, and 25 out of the 49 of the Volere-based classification schema.

As it can be seen in Table 7.4, five SRPs are situated below two classifiers: *Acceptance Tests*, *Analysis Stage Activities*, *Components History*, *Delivered Documents*, and *Project Progress Control*. The table also includes, for each NT-SRP, the *Entity Types*, the number of *Clusters* and *Extended Parts* of each SRP, as well as the number of *Parameters* for each part. The *Concept* for each SRP is not stated, since it is the same than the name of the SRP, neither the number of *Fixed Parts*, since it is the same than the number of clusters. Note that in the case of NT-SRPs, some SRPs to do not have any entity type (see Chapter 10 for further information). The NT-SRPs have among 1 and 2 clusters, each cluster has among 0 and 8 extended parts, and each part has at most 5 parameters.

The number of relationships found on the NT-SRP catalogue is shown in Table 7.5. The most frequent relationships are those ones among *SRP* and *SRP* (being the most common type *relates-to*), among *Domain* and *SRP* (being the most common type *requires*), and among *Parameter* and *Parameter* (being the most common type *supports*).

Table 7.4 – NT-SRP catalogue classified using the ISO/IEC 25010-based classification schema

ISO/IEC 25010 Classifiers		SRP	Entity Types	#Clusters / #Extended Parts	#Parameters
Functional Suitability	Functional Completeness	-	-	-	-
	Functional Correctness	-	-	-	-
	Functional Appropriateness	-	-	-	-
Perfor-mance	Time Behaviour	-	-	-	-
	Resource Utilization	-	-	-	-
	Capacity	-	-	-	-
Compa-tibility	Co-existence	-	-	-	-
	Interoperability	-	-	-	-
Usability	Appropriateness Recognisability	-	-	-	-
	Learnability	Document Characteristics	Document Type	2 Cl / (5 EP, 5 EP)	(1, 1, 1, 1, 1, 1) (2, 2, 2, 2, 2, 2)
		Delivered Documents (r)	Subsystem	1 Cl / (4 EP)	(2, 3, 4, 4, 3)
	Operability	-	-	-	-
	User Error Protection	-	-	-	-
	User Interface Aesthetics	-	-	-	-
	Accessibility	-	-	-	-
Reliability	Maturity	-	-	-	-
	Availability	-	-	-	-
	Fault Tolerance	-	-	-	-
	Recoverability	-	-	-	-
		-	-	-	-
Security	Confidentiality	-	-	-	-
	Integrity	-	-	-	-
	Non-repudiated	-	-	-	-
	Accountability	-	-	-	-
	Authenticity	-	-	-	-
Maintai-nability	Modularity	*Components History (r)*	Component Type	1 Cl (3 EP)	(2, 1, 1, 1)
	Reusability	-	-	-	-
	Analysability	*Analysis Stage Activities (r)*	N/A	1 Cl (3 EP)	(0, 1, 1, 1)
	Modifiability	-	-	-	-
	Testability	*Acceptance Tests (r)*	Subsystem	1 Cl (8 EP)	(2, 4, 4, 3, 3, 3, 2, 2)
Porta-bility	Adaptability	-	-	-	-
	Installability	-	-	-	-
	Replaceability	-	-	-	-
Business	Licensing Schema	Source Code Licenses	Subsystem	2 Cl / (0 EP, 0 EP)	(2) (2)
	Ownership	Intellectual Property Rights	Project Assets	1 Cl / (2 EP)	(2, 3, 1)
	Guarantees	Warranty	N/A	1 Cl / (6 EP)	(0, 0, 1, 4, 0, 1, 0)
	Costs	Cost Breakdown Structure	N/A	1 Cl / (2 EP)	(1, 1, 1)

	History	Community Support	N/A	1 CI (2 EP)	(0, 2, 0)
		Components History (r)	Comp. Type	1 CI (3 EP)	(2, 1, 1, 1)
Product	Deliverables	*Delivered Documents (r)*	Subsystem	1 CI / (4 EP)	(2, 3, 4, 4, 3)
		Source Code Documentation	Subsystem	1 CI / (3 EP)	(1, 1, 1, 2)
	Parameterization and Customization	-	-	-	-
Project	Business Scheduling	*Acceptance Tests (r)*	Subsystem	1 CI (8 EP)	(2, 4, 4, 3, 3, 3, 3, 2, 2)
		Analysis Stage Activities (r)	N/A	1 CI (3 EP)	(0, 1, 1, 1)
		Data Migration Activities	N/A	1 CI (5 EP)	(0, 0, 0, 0, 2, 1)
		Development Activities	N/A	1 CI (3 EP)	(0, 3, 2, 1)
		Final Acceptance	Subsystem	1 CI (1 EP)	(2, 5)
		Project Management Method	N/A	2 CI (2 EP, 2 EP)	(0, 0, 1) (0, 0, 1)
		Project Progress Control (r)	N/A	1 CI (1 EP)	(0, 1)
		Release	N/A	2 CI (3 EP, 3 EP)	(0, 2, 2, 0) (0, 0, 4, 4)
		System Implementation Scheduling	N/A	1 CI (5 EP)	(1, 1, 3, 3, 1, 2)
	Supplier Relationships	Access to Customer Premises	N/A	1 CI (3 EP)	(1, 0, 3, 1)
		Crash Response	Subsystem	1 CI (2 EP)	(1, 4, 3)
		Help Desk	Subsystem	1 CI (6 EP)	(1, 1, 2, 2, 2, 3, 2)
		Payment Scheduling	N/A	1 CI (2 EP)	(0, 2, 2)
		Privacy	N/A	1 CI (4 EP)	(0, 0, 0, 0, 1)
		Project Progress Control (r)	N/A	1 CI (1 EP)	(0, 1)
		Quality Assessment	Project Deliverables	2 CI (2 EP, 2 EP)	(2, 1, 3) (1 ,2, 1)
		Settlement of Disputes	N/A	1 CI (1 EP)	(1, 0)
		Steering Committee	N/A	1 CI (6 EP)	(1, 0, 2, 1, 1, 3, 2)
		Supplier People Assigned to the Project	People Type	2 CI (2 EP, 2 EP)	(2, 2, 4) (1, 1, 3)
		Meetings Organization	Meeting Type	1 CI (6 EP)	(1, 0, 2, 1, 1, 3, 2)
Supplier	Organizational Structure	Supplier Administrative Information	N/A	1 CI (2 EP)	(0, 2, 1)
		Supplier History	N/A	1 CI (1 EP)	(0, 1)
		Supplier Organization	N/A	1 CI (0 EP)	(0)
	Positioning and Strength	Supplier Economic Information	N/A	2 CI (3 EP, 3 EP)	(0, 2, 2, 0) (0, 0, 4, 4)
		Supplier Workforce	N/A	2 CI (4 EP, 4 EP)	(0, 0, 3, 2, 3) (0, 0, 2, 1, 0)
	Reputation	Supplier Projects Experience	N/A	2 CI (2 EP, 3 EP)	(0, 3, 3) (0, 1, 2, 2)
		Supplier Quality Certification	N/A	2 CI (3 EP, 3 EP)	(0, 1, 0 ,1) (0, 0, 1, 1)
	Services Offered	Training	Subsystem	1 CI (4 EP)	(1, 2, 2, 2, 3)
	Support	Maintenance Procedures	N/A	1 CI (2 EP)	(0, 2, 1)
		Maintenance Types	Subsystem	1 CI (6 EP)	(2, 2, 4, 2, 2, 3, 4)
Functional Reqs.	Content Management Systems	-	-	-	-

Table 7.5 – Number of relationships of the NT-SRP catalogue classified per type and elements

Relationship Elements	Conflicts	Disjoint	Relates-to	Requires	Rule	Subset	Supports
SRP - SRP	-	-	55	2	-	-	-
Part - SRP	-	-	-	8	-	-	-
Domain - SRP	-	-	-	29	-	-	-
Param - Param	-	-	-	-	-	1	34

Table 7.6 shows an example of NT-SRP, the *Supplier Economic Information*. This SRP produces requirements related to the goal of *Assessing the economic situation of the supplier*. The two clusters of the SRP allow to assess the economic situation of the supplier's company either by gathering information about the economic situation or by setting pre-requisites on the matter.

Table 7.6 – Supplier Economic Information NT-SRP

SRP ***Supplier Economic Information***	**Goal**	Assessing the economic situation of the supplier.		
	Keywords	Supplier's company, Economic information, Economic prerequisite		
	Description	This pattern expresses the need of assessing the economic situation of the supplier's company either by gathering information or by setting prerequisites.		
	Question	Does the customer need just information of the economic situation of the supplier or should the supplier fulfil some requisites on the matter?		
	Authors	GESSI-LIST		
	Sources	Specialized literature, SRSs from LIST		
	Date	13 January 2016		
	Version	v.3		
	Entities	---		
	Relationships	- SRP/SRP relates-to relationship. The following SRPs may relate to each other: *Supplier Administrative Information*, *Supplier Economic Information*, *Supplier History*, *Supplier Organization*, *Supplier People Assigned to the Project*, *Supplier Projects Experience, Supplier Quality Certification, Supplier Workforce*		
SRP Cluster *Economic Situation Information*	**Description**	This cluster expresses the need of gathering information about the economic situation of the suppliers company.		
	Behaviour	- *Fixed part, Company Turnover, Company Net Income, Consortium Information*: can be applied at most once each.		
	Fixed Part	**Template**	*The supplier shall provide economic information of its company.*	
	Extended Part *Company Turnover*	**Template**	*The supplier shall provide information of its company's turnover on the last %amountOfTime% %timeUnit%. Notice that in case the supplier´s company is part of a consortium, the information required here shall be related only to the company itself.*	
		Parameter name	**Domain name**	**Domain type**
		amountOfTime	*AmountOfTime*	*Float inv > 0*
		timeUnit	*TimeUnitTypes*	*[Seconds, Minutes, Miliseconds, Years, Hours, Days]*
	Extended Part *Company Net Income*	**Template**	*The supplier shall provide information of its company's net income on the last %amountOfTime% %timeUnit%. Notice that in case the supplier´s company is part of a consortium, the information required here shall be related only to the company itself.*	
		Parameter name	**Domain name**	**Domain type**
		amountOfTime	*AmountOfTime*	*Float inv > 0*
		timeUnit	*TimeUnitTypes*	*[Seconds, Minutes, Miliseconds, Years, Hours, Days]*
	Extended Part *Consortium Information*	**Template**	*In case the supplier is part of a consortium, it shall provide the same required economic information established above for the whole consortium.*	

SRP Cluster *Economic Situation Prerequisites*				
	Description	This cluster expresses the customer's prerequisites regarding the economic situation of the supplier's company. It has two extensions that precise the kind of economic prerequisite to be fulfilled: turnover, and net income. It has a last extension that allows setting prerequisites about economic situation of the consortium in case the supplier belongs to a consortium.		
	Behaviour	- *Fixed part, Consortium Prerequisites*: can be applied at most once each. - *Minimum Company Turnover, Minimum Company Net Income*: can be applied more than once if they are applied with disjoint values of amount of *time* and *time unit.* - *Minimum Company Turnover, Minimum Company Net Income*: the value of the *currency unit* should be the same in all the application of both extended parts.		
	Fixed Part	**Template**	*The supplier shall fulfil some economic situation prerequisites.*	
	Extended Part *Company Turnover*	**Template**	*The supplier's company shall have a minimum turnover of %amount% %currencyUnit% on the last %amountOfTime% %timeUnit%. Notice that in case the supplier's company is part of a consortium, the minimum turnover required here shall be applied only to the company itself.*	
		Parameter name	**Domain name**	**Domain type**
		amount	*AmountOfCur-rency*	*Float inv > 0*
		currencyUnit	*CurrencyUnit-Type*	*[JPY, EUR, GBP, USD]*
		amountOfTime	*AmountOfTime*	*Float inv > 0*
		timeUnit	*TimeUnitTypes*	*[Seconds, Minutes, Miliseconds, Years, Hours, Days]*
	Extended Part *Company Net Income*	**Template**	*The supplier's company shall have a minimum net income of %amount% %currencyUnit% on the last %amountOfTime% %timeUnit%. Notice that in case the supplier's company is part of a consortium, the minimum net income required here shall be applied only to the company itself.*	
		Parameter name	**Domain name**	**Domain type**
		amount	*AmountOfCur-rency*	*Float inv > 0*
		currencyUnit	*CurrencyUnit-Type*	*[JPY, EUR, GBP, USD]*
		amountOfTime	*AmountOfTime*	*Float inv > 0*
		timeUnit	*TimeUnitTypes*	*[Seconds, Minutes, Miliseconds, Years, Hours, Days]*
	Extended Part *Consortium Prerequisites*	**Template**	*In case the supplier is part of a consortium, each company of the consortium shall fulfil the economic situation prerequisites established.*	

For the *Economic Situation Information* cluster of the example, the fixed part states the need for the supplier to provide its economic information, and the extended parts precise what kind of economic information shall be provided (turnover, and net income) over different periods of time, and the need to provide information of the economic situation of the supplier's consortium (in case the supplier belongs to a consortium). Its behaviour explains that the four parts can be applied at most once each. In the *Economic Situation Prerequisites* cluster, the fixed part states the need for the supplier to fulfil economic situation prerequisites, and the three extended parts provide the detailed prerequisites. Two extended parts precise the kind of economic prerequisite to be fulfilled (turnover, and net income), and the last one allows setting prerequisites about economic situation of the consortium in case the supplier belongs to a consortium. The behaviour states that the fixed part and the *Consortium Prerequisites* extended part can be applied at most once each, while the *Minimum Company Turnover* and the *Minimum Company Net Income* extended parts can be applied more than once if they are applied with disjoint values of amount of time and time unit, since it is possible to state different prerequisites for the supplier (i.e. money quantities) for different periods of time. The behaviour also states that the currency used in both extended parts should be the same in all their applications, since it does not make sense, for instance, to state prerequisites in Euros and in Dollars in the same project. The domains of the parameters in the first cluster are the amount of time and time unit for which the supplier shall provide the information. Apart from these two parameters, the second cluster also has as parameters the currency amount and currency type that state the prerequisites on the supplier's economic situation. The *Supplier Economic Information* SRP has a relationship of type *relates-to* with all the patterns that specify restrictions over the supplier: *Supplier Administrative Information*, *Supplier History*, *Supplier Organization*, *Supplier People Assigned to the Project*, *Supplier Projects Experience*, *Supplier Quality Certification*, and *Supplier Workforce*.

7.4 Functional SRP Catalogue

The F-SRP catalogue obtained from the 6 analysed SRS documents is composed of 44 SRPs for the CMS domain. Table 7.7 contains the F-SRP catalogue organized using the classification schema based on the ISO/IEC 25010 standard.

The characteristics of the ISO/IEC 25010 standard system and software quality model catalogue and the ISO/IEC standard 9126 software quality model catalogue include one characteristic related to the functionality of the systems, namely *Functionality Suitability* and *Functionality*, respectively. However, they are defined as "the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions", and as both standards state the characteristics are not directly concerned with the functional specification of a system. Therefore, in order to accommodate functional requirements in both classification schemas, it was necessary to add a new root classifier *Functional Requirements*. In the case of the Volere-based classification schema, the *Functional Requirements* root classifier already existed.

Table 7.7 – F-SRP catalogue classified using the ISO/IEC 25010-based classification schema*

ISO/IEC 25010 Classifiers				SRP	Entity Types	#Clusters / #Extended Parts	#Parameters
Functional Requirements	Content Management Systems	Users	Users Management	-	-	-	-
			Users actions	-	-	-	-
		Roles	Roles Management	-	-	-	-
		Groups	Groups Management	-	-	-	-
		Content	Contents Management	*Assets to Import/Export (r)*	Subsystem	2 Cl (5 EP, 4 EP)	(2, 3, 4, 4, 4) (2, 3, 3, 3, 3)
				Broadcast Features	Subsystem	1 Cl (11 EP)	(1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1)
				Content Annotation	Subsystem	2 Cl (7 EP, 7 EP)	(2, 3, 2, 3, 3, 3, 2, 2) (1, 2, 1, 2, 2, 2, 1, 1)
				Content Customization	Subsystem	2 Cl (2 EP, 2 EP)	(2, 3, 4) (1, 2, 3)
				Content Lifetime	Subsystem	2 Cl (4 EP, 4 EP)	(2, 3, 3, 4, 3) (1, 2, 2, 3, 2)
				Content Management	Subsystem	2 Cl (13 EP, 13 EP)	(2, 4, 2, 2, 2, 3, 3, 3, 3, 4, 3, 5, 3, 3) (1, 3, 1, 1, 2, 1, 2, 2, 2, 3, 2, 4, 2, 2)
				Content Metadata	Subsystem	2 Cl (4 EP, 4 EP)	(2, 3, 3, 4, 3) (1, 2, 2, 3, 2)
				Content Preview	Subsystem	2 Cl (3 EP, 3 EP)	(2, 3, 3, 3) (1, 2, 2, 2)
				External Broadcast	Subsystem	2 Cl (5 EP, 5 EP)	(2, 3, 3, 2, 2, 2) (1, 2, 2, 2, 2, 2)
				Internal Automatic Broadcast	Subsystem	2 Cl (4 EP, 4 EP)	(2, 3, 3, 3, 2) (1, 2, 2, 2, 1)
				Internal Manual Broadcast	Subsystem	2 Cl (3 EP, 3 EP)	(2, 2, 3, 2) (1, 1, 2, 1)
				Multilingualism	Subsystem	1 Cl (6 EP)	(2, 2, 2, 3, 2, 1, 2)
			Contents Security	-	-	-	-
			Contents Ve-rifiabiliness	Content Version Management	Subsystem	2 Cl (4 EP, 4 EP)	(1, 1, 1, 2, 1) (2, 2, 2, 3, 2)
				History	Subsystem	1 Cl (4 EP)	(1, 2, 3, 3, 2)
		Folders	Folders Management	*Storage Hierarchy (r)*	Subsystem	2 Cl (6 EP, 6 EP)	(2, 3, 3, 4, 3, 4, 2) (1, 2, 2, 3, 2, 3, 1)
				URL Features	Subsystem	1 Cl (12 EP)	(2, 3, 2, 3, 2, 2, 3, 3, 2, 4, 3, 2, 2)
			Folders Security	-	-	-	-
		Alias	Alias Management	External Content References	Subsystem	1 Cl (3 EP)	(1, 2, 2, 2)
				Websites with External Content	Subsystem	2 Cl (2 EP, 2 EP)	(2, 4, 4) (1, 3, 3)
			Alias Security	-	-	-	-
		Query	Searches	Indexing	Subsystem	1 Cl (8 EP)	(1,3,3,3,3,3,3,3,1)
				Search Engine	Subsystem	1 Cl (12 EP)	(1, 2, 2, 3, 3, 3, 4, 3, 3, 3, 3, 3, 3)
				Statistics Generation	Subsystem	1 Cl (3 EP)	(1, 2, 5, 3)

			Search API	-	-	-	-
			Search Security	-	-	-	-
		Lifecycle	Lifecycle Management	Pre-Publication Actions	Subsystem	2 Cl (3 EP, 3 EP)	(2, 3, 4, 3) (1, 2, 3, 2)
				Post-Publication Actions	Subsystem	2 Cl (2 EP, 1 EP)	(2, 3, 4) (2, 3)
				Publication Workflow	Subsystem	2 Cl (17 EP, 17 EP)	(2, 4, 3, 2, 3, 2, 3,3, 2, 2, 2, 3, 3, 3, 3, 2, 2, 2) (1, 3, 2, 1, 2, 1, 1,2, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1)
				Workflow Management	Subsystem	2 Cl (6 EP, 6 EP)	(2, 3, 3, 2, 2, 2, 2) (1, 2, 2, 2, 1, 1, 1)
			Lifecycle Security	-	-	-	-
		Mail	Mail Files Management	*Assets to Import/Export (r)*	Subsystem	2 Cl (5 EP, 4 EP)	(2, 3, 4, 4, 4, 4) (2, 3, 3, 3, 3)
				E-mail Notifications (r)	Subsystem	1 Cl (3 EP)	(1, 2, 3, 3)
				Mailing Lists (r)	Subsystem	1 Cl (4 EP)	(1, 2, 2, 2, 2)
			Mail Security	-	-	-	-
		Web Content	Web Content Management	Remote Access	Subsystem	1 Cl (2 EP)	(1, 2, 2)
				Website Features	Subsystem	1 Cl (8 EP)	(2, 2, 2, 2, 3, 4,4, 4, 4)
				Website Management	Subsystem	1 Cl (21 EP)	(1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 2, 2, 3, 3, 1)
				Website Naviga-tion Features	Subsystem	1 Cl (4 EP)	(2, 3, 2, 2, 1)
				Website Users	Subsystem	1 Cl (18 EP)	(2, 3, 2, 3, 3, 2, 3, 3, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2)
			Web Content Security	-	-	-	-
		Media	Media Management	Buyable Features	Subsystem	1 Cl (6 EP)	(1, 2, 3, 3, 1, 2, 1)
				Electronic Payment	Subsystem	1 Cl (1 EP)	(1, 2)
				Newsletters	Subsystem	1 Cl (14 EP)	(1, 2, 2, 2, 2, 2, 3, 2, 2, 1, 2, 2, 2, 1, 2)
		Agenda	Agenda Management	Automatic Incoming Faxes Management	Subsystem	1 Cl (4 EP)	(1, 2, 1, 2, 2)
				Contacts Management	Subsystem	2 Cl (2 EP, 2 EP)	(2, 2, 1) (1, 3, 2)
				E-mail Notifications (r)	Subsystem	1 Cl (3 EP)	(1, 2, 3, 3)
				Mailing Lists (r)	Subsystem	1 Cl (4 EP)	(1, 2, 2, 2, 2)
				To-Do List	Subsystem	2 Cl (5 EP, 5 EP)	(2, 3, 2, 3, 2, 3) (1, 2, 1, 2, 2, 2)
		Storage	Storage Management	Automatic Storage	Subsystem	2 Cl (3 EP, 3 EP)	(2, 3, 2, 2) (1, 2, 1, 1)
				Storage Compression	Subsystem	3 Cl (3 EP, 3 EP, 3 EP)	(1, 1, 1, 1) (2, 1, 1, 1) (2, 1, 1, 1)
				Storage Features	Subsystem	2 Cl (7 EP, 7 EP)	(2, 3, 3, 4, 3, 4, 3, 2) (1, 2, 2, 3, 2, 3, 2, 1)
				Storage Hierarchy (r)	Subsystem	2 Cl (6 EP, 6 EP)	(2, 3, 3, 4, 3, 4, 2) (1, 2, 2, 3, 2, 3, 1)
		Text Input	Text Input Management	Editor	Subsystem	1 Cl (10 EP)	(1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2)
				Formularies Features	Subsystem	1 Cl (7 EP)	(1, 2, 3, 2, 2, 2, 1, 2)
				Spell-Checker	Subsystem	1 Cl (2 EP)	(1, 2, 2)

* The classifiers shown in this table are only those ones related to functional requirements. For the complete list of classifiers of this classification schems see Section 7.2 or Section 7.3.

In three schemas that are being used, the *Functional Requirements* classifier will be decomposed in as much domains as needed. In this case, a compound classifier *Content Management System* was added to the root. As starting point for the decomposition of *Content Management System*, the ISO/IEC-9126-1-based quality model for the CMS domain (proposed by Franch et al. [177]) was used. Although, some changes were needed in order to accommodate some of the F-SRPs. First, it was necessary to add four new subcharacteristics as direct children of *Content Management System*, regarding the management of *Media*, *Agenda*, *Storage* and *Text Input*. Second, it was necessary to add a new subcharacteristic *Content Verifiabiliness* as child of the existing *Content* subcharacteristic.

The coverage of the catalogue for the ISO/IEC 25010-based, the ISO/IEC 9126-based and the Volere-based classification schemas is the same. It covers 1 characteristic of the first level (*Functional Requirements*), 1 subcharacteristic of the second level (*Component Management Systems*), 11 out of 14 subcharacteristics of the third level, and 12 out of 24 subcharacteristics of the fourth level.

As it can be seen in Table 7.7, four SRPs are situated below two classifiers: *Assets to import/export*, *E-mail notifications*, *Mailing Lists* and *Storage Hierarchy*. The table also includes, for each F-SRP, the *Entity Types*, the number of *Clusters* and *Extended Parts* of each SRP, as well as the number of *Parameters* for each part. The *Concept* for each SRP is not stated, since it is the same than the name of the SRP, neither the number of *Fixed Parts*, since it is the same than the number of clusters. Note that the SRPs have among 1 and 3 clusters, each cluster has among 1 and 21 extended parts, and each part has at least 0 and at most 5 parameters.

The number of relationships found on the F-SRP catalogue is shown in Table 7.8. The most frequent relationships are those ones among *SRP* and *SRP*, and among *Parameter* and *Parameter*. In both cases, the most common type is *supports*.

Table 7.8 – Number of relationships of the F-SRP catalogue classified per type and elements

Relationship Elements	Conflicts	Disjoint	Relates-to	Requires	Rule	Subset	Supports
SRP - SRP	-	-	64	6	-	-	559
Cluster - Cluster	-	-	-	10	-	-	-
Part - SRP	-	-	-	1	-	-	-
Part - Cluster	-	-	-	6	-	-	-
Param - Param	-	-	-	-	-	22	168

An example of F-SRP is shown in Table 7.9, namely the *Content Version Management*. This SRP produces requirements related to the goal of *Being able to manage versions*. This goal can be attained by having version management over all contents stored in the system (*Homogeneous Version Management* cluster), or by having version management over specific contents stored in the system (*Heterogeneous Version Management* cluster). In the *Homogeneous Version Management* cluster, the fixed part states that management version shall be carried out over the stored contents, and the four extended parts express specific requirements on the need of: having automatic versioning (*Automatic Versions*), retrieving old versions (*Version Retrieval*), numbering the versions with a specific mode (*Version Numbering*) and saving contents as a previous version *(Savings as Previous Versions*). Its behaviour explains that the four parts can be applied at most once each for each subsystem. For the *Heterogeneous Version Management* cluster, the fixed part states the contents on which versioning will be implemented, and the four extended parts express the same specific requirements that the extended parts of the other cluster, but detailing the specific contents for which the requirements apply. The behaviour states that all the parts (except *Specific Version Numbering*) can be applied at most once each for each subsystem; *Specific Version Numbering* can be applied more than once provided it is applied for disjoint pairs of subsystem and content types, since it might be possible that in a specific subsystem, some contents require a type of numbering, and some other contents a different type. The domains of the parameters of both clusters correspond to the set of possible subsystems and numbering modes. In addition, the second cluster has a parameter that corresponds to the set of possible contents.

The *Content Version Management* SRP is involved in three relationships:

- The *Content Version Management* SRP may require *Contents Management* SRP, since it does not make sense to manage versions of contents if it is not possible to manage contents.
- The *Content Version Management* SRP relates-to the following SRPs: *Content Annotation*, *Content Customization*, *Content Lifetime*, *Content Metadata*, and *Content Preview*. The reasoning behind this is that, as the *Content Version Management* SRP defines requirements over the contents stored in the system, if a customer is interested in this pattern, the customer might also be interested in any of the other patterns that set conditions over the contents.

- The specific contents defined by the *Heterogeneous Version Management* cluster of the *Content Version Management* SRP must be a subset of the ones defined by the *Heterogeneous Version Management* cluster of the *Content Management* SRP, since it does not make sense to manage versions of specific contents if it is not possible to manage them.

Table 7.9 – Content Version Management F-SRP

SRP *Content Version Management*	**Goal**	Being able to manage versions		
	Keywords	Version, Management, Automatic versioning, Version retrieval, Content, Content type		
	Description	This pattern expresses the need of having a system that manages versions of contents.		
	Question	Does the customer require different types of version management depending on the type of content?		
	Authors	GESSI-LIST		
	Sources	Specialized literature, SRSs from LIST		
	Date	13 January 2016		
	Version	v.3		
	Entities	Subsystem		
	Relationships	- SRP/SRP require relationship. *Content Version Management* SRP may require *Contents Management* SRP. - SRP/SRP relates-to relationship. *Content Version Management* SRP relates-to the following SRPs: *Content Annotation*, Content Customization, *Content Lifetime*, *Content Metadata*, *Content Preview*. - Parameter/Parameter subset relationship. *contentTypes* parameter of the applications of *Heterogeneous Version Management* cluster (*Content Version Management* SRP) must be a subset of the ones used in the applications of the *Heterogeneous Content Management* cluster (*Content Management* SRP).		
SRP Cluster *Homogeneous Version Management*	**Description**	This cluster establishes the need of having a version management over all contents stored in the system. Specifically, the cluster allows to define the possibility of having automatic versioning, the possibility to recover a previous version, the possibility to number the versions automatically or manually, and the necessity of not allowing to save as a previous version.		
	Behaviour	- *Fixed Part, Automatic Versions, Version Retrieval, Version Numbering, Savings as Previous Versions*: can be applied at most once each for a same *subsystem*.		
	Fixed Part	**Template**	*The %subsystem% shall manage version of the stored contents.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
	Extended Part *Automatic Versions*	**Template**	*The %subsystem% shall propose automatically the creation of new versions depending on the changes done in the content.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
	Extended Part *Version Retrieval*	**Template**	*The %subsystem% shall allow the retrieval of a previous version of a content.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
	Extended Part *Version Numbering*	**Template**	*The %subsystem% shall allow the numbering of versions of a content to be %numberingMode%.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
		numberingMode	*NumberingMode*	*[Automatic, Manual, ...]*
	Extended Part *Savings as Previous Versions*	**Template**	*The %subsystem% shall not allow saving a new version of content as it was a previous one.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*

SRP Cluster	Part			
Heterogeneous Version Management	Description	This cluster stablishes the need of having a version management over specific contents stored in the system. Specifically, the cluster allows to define what contents shall be versioned, the possibility of having automatic versioning, the possibility to recover a previous version, the possibility to number the versions automatically or manually, and the necessity of not allowing to save as a previous version.		
	Behaviour	- *Fixed Part*, *Specific Automatic Versions*, *Specific Version Retrieval*, *Specific Savings as Previous Versions*: can be applied at most once each for a same *subsystem*. - *Specific Version Numbering*: can be applied at most once for each pair of *subsystem* and *content types*. - *Specific Automatic Versions*, *Specific Version Retrieval*, *Specific Version Numbering*, *Specific Savings as Previous Versions*: should use a subset of the *content types* used in *Fixed Part's* applications for the same subsystem.		
	Fixed Part	**Template**	*The %subsystem% shall do versioning over %contentTypes% contents.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
		contentTypes	*ContentTypes*	*Set(ContentType = [publications, reports, base documents, etc.])*
	Extended Part *Specific Automatic Versions*	**Template**	*The %subsytem% shall propose automatically the creation of new versions over %contentTypes% depending on the changes done in the content.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
		contentTypes	*ContentTypes*	*Set(ContentType = [publications, reports, base documents, etc.])*
	Extended Part *Specific Version Retrieval*	**Template**	*The %subsystem% shall allow the retrieval of a previous version of %contentTypes% contents.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
		contentTypes	*ContentTypes*	*Set(ContentType = [publications, reports, base documents, etc.])*
	Extended Part *Specific Version Numbering*	**Template**	*The %subsystem% shall allow the numbering of versions of a %contentType% content to be %numberingMode%.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
		contentTypes	*ContentTypes*	*Set(ContentType = [publications, reports, base documents, etc.])*
		numberingMode	*NumberingMode*	*[Automatic, Manual, ...]*
	Extended Part *Specific Savings as Previous Versions*	**Template**	*The %subsystem% shall not allow saving a new version of %contentTypes% contents as if it was a previous one.*	
		Parameter name	**Domain name**	**Domain type**
		subsystem	*Subsystem*	*[System, Server system, Client system, ...]*
		contentTypes	*ContentTypes*	*Set(ContentType = [publications, reports, base documents, etc.])*

CHAPTER 8 Return-On-Investment of an SRP Catalogue

8.1 Introduction

Although the adoption of requirements reuse might have plenty of benefits for an organization, as stated by Wiegers and Beatty [26], it also implies several challenges, among them, the need for an initial investment and the maintenance of the reusable knowledge. Hence, in order to reuse requirements, organizations face a fundamental question: *Is it worth to invest on the adoption of a requirements reuse strategy?*. Introducing requirements reuse involves making a decision of a greater degree than considering only the benefits, and it should also include productivity issues, which are actually measured in terms of effort, cost, and economic benefits.

Therefore, organizations need to ensure the feasibility of adopting requirements reuse by assessing their goals, the resources they can invest and the expected benefits. In that sense, reuse economic models try to give an economic value to activities where reuse is involved, putting into numbers the significant contribution and competitive advantage that reuse can give to an organization.

In spite of the benefits of having reuse economic models, when reviewing the existing approaches for economic models, none of them deal with reuse at the requirements level, and even less when reuse is achieved by using an approach similar to SRPs (see Section 8.2). Thus, the adoption of requirements reuse is usually made without evaluating their economic impact. To make informed decisions, it is necessary to know how many instantiations (i.e. applications) are needed before savings pay off for the up-front investment in building a requirements reuse base of knowledge.

Then, in order to measure the economic benefits that the use of the PABRE framework can bring to an organization, it is necessary to create a reuse economic model for the specific SRP case (presented in Section 8.3). However, it is worth noting that the proposed model can be easily generalized to other requirements reuse techniques, and that it faces some limitations (Section 8.4).

8.2 Background

Current research on requirements reuse has little support to analyse the cost and benefits of requirements reuse based on economics. There exist a few approaches to cost models for software reuse in general that state requirements reuse should be incorporated into cost models, as the ones proposed by Barns et al. [178] and Poulin [179], but they do not quantify it. Other works quantify the level of reuse, but no the economics benefits or costs associated to it, e.g. the work of Daneva [180]. As far as the applicant knows, only Eriksson's et al. [50] and Goldin and Berry's [57] studies quantify the level of requirement reuse and the economic saving that it entails, but they do not propose a model that allows to measure it. Requirements reuse involves fundamental assumptions that economic methods need to reflect, especially when it comes to defining potential economic benefits and the payback time.

By contrast, there exist several approaches that propose metrics for estimating cost savings in software development and maintenance. For instance, Poulin [181] considers the reuse of assets in developing individual applications, and the potential costs savings that it implies to the development and maintenance. Mac-Cormack et al. [182] extracts coupling metrics using Dependency Structure Matrices (DSMs) for inferring the likelihood of change propagation and, hence, future maintenance costs. Baldwin and Clark [183] presents a generic expression for evaluating the option to redesign a module also based on DSMs that is used by Carriere et al. [184] and Sullivan et al. [185]. Other models propose metrics for factors usually not incorporated in the previous models such as: the reused knowledge that modify the source of reuse, as Bollinger and Pfleeger's [186] and Nazareth and Rothenger's [187] models; the need of defining parts that are not obtained from the reuse knowledge base, as Rothenberger and Dooley's [188] model; and the quality gains as consequence of reuse, as Frakes and Terry's [189] and Mili's et al. [190] models.

An inspiring area for economic evaluation is Software Product Lines (SPLs). A survey by Ali et al. [191] summarizes twelve economic models for SPLs. The SIMPLE approach (proposed by Clements et al. [192]) and Poulin's approach [193] are some of the most widespread economic models for SPLs. SIMPLE comprises a set of cost functions that can be used to construct equations that can answer a number of questions such as whether the SPL approach is the best option for development and what is the ROI for this approach. Specifically, SIMPLE comprises a set of seven cost factors:

- C_{org}, upfront investments to establish a SPL infrastructure.
- C_{cab}, the cost to build reusable assets of a SPL.
- C_{unique}, the cost to develop unique parts of products in a SPL.
- C_{reuse}, the cost of reusing reusable assets in a product inside a SPL.
- C_{cabu}, the cost to evolve the core asset in a SPL.
- C_{prod}, the cost to build a product in a stand-alone fashion.
- C_{evo}, the cost to evolve a product in a stand-alone fashion.

On the other hand, Poulin bases his reuse-based model in two parameters:

- RCR (Relative Cost of Reuse). Assuming that the cost to develop a reusable asset equals one unit of effort, RCR is the portion of this effort that it takes to reuse a reusable asset without modification (black-box reuse).
- RCWR (Relative Cost of Writing for Reuse). Assuming that the cost to develop a new asset for one-time use equals one unit of effort, RCWR is the portion of this effort that it takes to write a similar reusable asset.

Later on, these parameters are used to calculate the benefits associated to reuse using the Reuse Cost Avoidance (RCA), which is further decomposed into the Development Cost Avoidance (DCA) and the Service Cost Avoidance (SCA), and the cost associated to reuse through the Cost of Developing Common Software (CSW_{dev_costs}) and the Cost of Maintaining this Common Software ($CSW_{service\_costs}$).

Another inspiring area for economic evaluation is ROI models. Among the different techniques that allow to quantify costs and benefits (such as technical debt, fully described by Letouzey [194]), ROI models hold a prominent position. Examples of that is the big amount of works proposing that kind of models, e.g. the proposals of Gaffney and Cruickshank [195], Rine and Nada [112], Erdogmus et al. [196], Van Solingen [197], and Biffl et al. [198]. These models seek to determine if a reuse investment will pay off in the future, analysing the net benefit of reuse after expending some level of resources. As expected, some of these models are also focused on SPLs, such as the one proposed by Böckle et al. [199]. There are also some works that deal with the ROI of requirements management tools (e.g. Denney's [200] and Moksony's [201] works), but they focus on whether it is worth to invest in such a tool and do not deal with whether requirements reuse pays off.

Despite the existence of this inspiring body of work from SPLs and ROI, the proposed models are not directly applicable to requirements reuse, and therefore neither to SRPs. Although SPLs and SRPs have similarities (both have reuse as their core strategy) they have also an important difference: as Decker et al. states [202], on SPLs, requirements are just one asset that may be reused, while SRPs are only focused on requirements.

Up to the applicant's knowledge, there is no specific economic model for estimating whether it is worth or not to invest in a requirements reuse strategy for an organization, and even less if reuse is achieved by means of requirement patterns. Due to the lack of research in this specific area, this thesis aims at adopting and adapting existing results in economic models for SPLs, and more generic metrics about cost savings.

8.3 Definition of the ROI Model for SRPs

The SRP reuse economic model is based on the ROI formula proposed by Biffl et al. [198]. In the SRP case, the benefits are the improvements in the elicitation and specification process and the non-done errors because of the use of SRPs, and the costs are the expenses of constructing the SRP catalogue, managing it, and creating the unique parts of SRSs (i.e. the requirements that are unique in a software project):

$$ROI = \frac{Benefits - Costs}{Costs}$$

For the decomposition of *Costs* and *Benefits*, six cost-benefit factors have been identified for the SRP adoption. The formulation of these factors started by adopting Poulin's method [193] for measuring reuse in SPLs. Poulin's approach was chosen over Clements' et al. SIMPLE approach [192] since it has been applied in industry and offers parameters to operationalize it, and SIMPLE includes more cost factors than needed for the SRP case (such as C_{cabu}) and the functions to calculate the benefits are not stated, but just introduced as something that should be taken into account.

For the benefit factors, the SRP ROI model adapts the Development Cost Avoidance (DCA) and the Service Cost Avoidance (SCA) presented by Poulin [193]. In addition, the model adopt and adapt for SRPs the defined Common Software Development Costs (CSW_{dev_costs}) and Common Software Service Costs ($CSW_{service\_costs}$) cost factors.

To complete the model, some further aspects not considered by Poulin's method have been explored. First, as stated for software reuse in general by Bollinger and Pfleeger [186] and Nazareth and Rothenger [187], it is important to consider the assets that have been reused modifying the knowledge source. For SRPs, this corresponds to the requirements that come from applying an SRP but that do small variations in the text of the templates (never being these variations substantial changes). Second, it is necessary to take into account the changes done into the SRP catalogue and, therefore, evolution, as proposed with the propagation cost metric by MacCormack et al. [182] and Clements et al. [192]. Third, as proposed by Rothenberger and Dooley [188] and Clements et al. [192], the unique elicitation costs have been added, i.e. the cost to develop the unique parts of an SRS that cannot be elicited using SRPs (it embraces the new requirements created from scratch because either there is no SRP that accomplish the requirement's goal, an SRP cluster is missing in an SRP or a part is missing in an SRP cluster). Finally, it has to be considered if reuse implies a gain in quality, and if this is the case, if it also implies an economical benefit that has to be incorporated into the model, as presented by Frakes and Terry [189] and Mili et al. [190]. In the SRP economic model, the quality gain is not considered as an independent economic benefit, but it is implicitly incorporated in the figure that expresses the errors no committed because SRPs are used (i.e. using SRPs implies more quality in the requirements which at the end is reflected in less errors).

The following list contains the six cost-benefit factors identified for the SRP ROI model. The former two factors identify the benefits and the later ones the costs:

- ESCA, Elicitation and Specification Cost Avoidance, similar to Poulin's DCA [181] [193]. It is the benefit from reusing SRPs in SRSs compared to building them from scratch, taking into account both the requirements that are reused without or with small SRP templates' modifications.
- SCA, Service Cost Avoidance, as presented in Poulin's work [181] [193]. They are the benefits coming from the avoided errors in SRSs because of the use of SRPs. These avoided errors come from the requirements that are reused without modifications, but also from the ones that are reused doing small variations of the SRP templates.

- AESC, Additional Elicitation and Specification Costs, similar to Poulin's Common Software Development Costs (CSW_{dev_costs}) [193]. It is the cost of the initial investment, i.e. developing an SRP catalogue.
- AMC, Additional Maintenance Costs, similar to Poulin's Common Software Service Costs ($CSW_{service\_costs}$) [193]. Cost coming from solving the errors in the SRP catalogue.
- CUC, Catalogue Update Costs. It is the cost of updating the SRP catalogue (adapted from MacCormack's et al. proposal [182]). It embraces the cost of adding SRPs (or an SRP cluster or part) to the catalogue, the cost of modifying existing SRPs, the cost of propagating the new additions or modifications (e.g. adding or modifying SRPs relationships), and finally the cost of deleting existing SRPs.
- UESC, Unique Elicitation and Specification Costs. It is the cost to develop the unique parts of an SRS that cannot be elicited and specified using SRPs (adapted from Rothenberger and Dooley's work [188]). In the SRPs case, these unique parts correspond to requirements created from scratch because either there is no SRP that accomplish the requirement's goal, or because an SRP cluster is missing in an SRP or a part is missing in an SRP cluster.

Putting everything together, given a number *n* of SRSs built on top of an SRP catalogue, and a number *m* of SRPs (the number of SRPs in a catalogue), the benefits and costs of adopting SRPs are defined as:

$$Benefits = \sum_{i=1}^{n}(ESCA_i + SCA_i)$$

$$Costs = AESC + AMC + CUC + \sum_{i=1}^{n}(UESC_i)$$

Table 8.1 shows the formulas to calculate the six cost-benefit factors previously defined. The twelve basic parameters that are required for calculating the six cost-benefit factors are presented in Table 8.2.

Table 8.1 – Cost-benefit factors to calculate the ROI of adopting an SRP catalogue in an organization

Factor	Description of the cost-benefit factors (adapted for the SRP context)
ESCA	Elicitation and Specification Cost Avoidance: Benefits from reusing SRPs [181] [193]. In the SRP case, it also takes into account the requirements that have been reused with modification [186] [187]. $ESCA = (RR\ x\ (1 - RCR)\ x\ RC) + (RR_{mod}\ x\ (1 - RCR_{mod})\ x\ RC)$
SCA	Service Cost Avoidance: Benefits coming from the avoided errors because of the use of SRPs [181] [193]. $SCA = (RR + RR_{mod})\ x\ ER\ x\ EC$

AESC	Additional Elicitation and Specification Costs: Cost of developing an SRP catalogue [193]. $AESC = RCWR\ x \sum_{j=1}^{m}(R_{SRPj})\ x\ RC$
AMC	Additional Maintenance Costs: Costs of fixing errors in an SRP catalogue [193]. $AMC = \sum_{j=1}^{m}(R_{SRPj})\ x\ ER\ x\ EC$
CUC	Catalogue Update Costs: Costs of changing or adding SRPs to a catalogue [182]. $CUC = Update\ Development + Update\ Maintenance + Propagation =$ $= \left(R_{SRP\_UPD_j}\ x\ RCWR\ x\ RC\right) + \left(R_{SRP\_UPD_j}\ x\ ER\ x\ EC\right) + \left(\left(\sum_{j=1}^{m} R_{SRPj}\right) x\ RC\ x\ PR\right)$
UESC	Unique Elicitation and Specification Costs: Costs to develop the unique parts of an SRS [188]. $UESC = R_{new}\ x\ RC$

Table 8.2 – Basic parameters in order to feed the factors of Table 8.1

Parameter	Description of the parameters (adapted for the SRP context)
RCR	Relative Cost of Reuse: Amount of effort to locate, evaluate and integrate a requirement in an SRS coming from an SRP without modification rather than elicit and specify it from scratch (i.e. without using SRPs) [181].
RCR_{mod}	Relative Cost of Reuse with Modification: Amount of effort to locate, evaluate and integrate a requirement in an SRS coming from modifying an SRP rather than elicit and specify it from scratch (i.e. without using SRPs) (adapted from [186] [187]).
RCWR	Relative Cost of Writing for Reuse: Ratio of the portion of effort that takes to specify a requirement for reuse relative to the cost of specifying it for one-time use [193].
EC	Error Cost: Average cost of solving an error in an SRS, in euros per error [181] [193].
RC	Requirement Cost: Cost of eliciting and specifying a requirement from scratch (i.e. without using SRPs) [181] [193].
PR	Propagation Rate: Percentage of the requirements affected in the SRP catalogue when performing updates (e.g. removing SRPs or adding SRPs with new dependencies) (adapted from [182]).
ER	Error Rate: Historical rate of errors in an SRS [181] [193].
RR	Reused Requirements: Number of requirements in a project that come as a direct application (without modification) of SRPs [181].
RR_{mod}	Reused Requirements with Modification: Number of requirements in a project that come as an indirect application (with modification) of SRPs (adapted from [187] [186]). Such modifications are just small variations of the templates of the SRPs, but never substantial changes.
R_{new}	New Requirements: Number of new requirements (i.e. not elicited and specified using SRPs) in a project [193].
R_{SRP}	Requirements in an SRP: Number of requirements in an SRP [193].
R_{SRP_UPD}	Requirements in an SRP that are updated: Number of requirements in an SRP that are modified, removed or added to the catalogue (adapted from [182]).

8.4 Discussion

The presented economic model allows quantifying the value that the adoption of SRPs brings to an organization. It translates measured or estimated data (i.e. metrics) into monetary terms (i.e. cost-benefit analysis). Then, it is used as the basis for analysing the economic value of an SRP catalogue (i.e. valuation) that is adapted by an organization in the pursuit of its business strategies. Therefore, the work aligns with Erdogmus' et al. vision on economic activities in software industry, that fall into 4 levels: metrics, cost-benefit analysis, valuation and business strategy [196].

The proposed model could be easily generalized to any other requirements reuse technique, especially if it consists on a set of requirements stored in a repository or a similar resource (no matter how they are organized or accessed). The most important change is adapting the parameters of Table 8.2 that take into account SRPs (RCR, RCR_{mod}, PR, RR_{mod}, R_{SRP} and R_{SRP_UPD}) and adjust them to the new requirements reuse approach. In addition, the summations for R_{SRPj} might need to be adapted or removed, depending on the approach.

The strongest points of the SRP economic model are:

- It translates SRPs costs and benefits into monetary values, which can be considered an innovative approach in requirements reuse research and practice.
- The integration of different metrics from existing models that complement each other evaluating several SRP-relevant aspects.
- It provides guidelines for easily collecting and reporting data for practitioners, and for using it to make a business case.

On the other hand, potential weaknesses of this approach are:

- It does not consider SRPs elements degeneration over time, as Ganesan et al. suggests [203]. Therefore, aspects like the decay of the value of knowledge as technology and software domains evolve or new projects being so different that it is not possible to reuse information from the old ones are not taken into account by the model.
- The risk if inaccuracy increases when neither real nor historical data are available or updated. When the economic model is used to predict the ROI of a completely new SRP reuse adoption in an organization, there is no real data for giving value to some of the parameters (e.g. RCWR or EC) since the data does not exist yet. In this case, the accuracy of the model totally depends on expert intuition and historical data. In addition, historical data must be continuously updated, since some values of effort-related parameters (such as RCR) are expected to decrease each time a requirements analysts applies an SRP.

However, it is important to highlight that it is planned to enrich the economic model by adding more metrics, such as technical debt [194], degeneration over time [203], risk metrics [196] [198], and indirect benefits [204].

In order to validate the SRP ROI model, it is important that an organization would have already adopted SRPs and that it would be willing to adopt them and to use them for more than one project, since the validation requires data of the use of SRPs in more than one project. As it is less probable that an organization has already adopted and used SRPs, the best idea to validate the economic model is to carry out a case study that will encompass different IT projects of the same organization. In this case study, the organization will use the PABRE framework to elicit and specify requirements, thus being able to check the appropriateness of the economic model with the data gathered in these projects.

CHAPTER 9 The PABRE System

9.1 Architecture

The PABRE System (Figure 9.1) is the technological support of the PABRE framework, helping requirements analysts and RE experts to use, maintain and update the SRP catalogue. It is composed of three subsystems: PABRE-Man, PABRE-Proj and PABRE-WS. The use case diagram of the system is shown in Figure 9.2.

Requirements analysts may use the SRP catalogue through the PABRE-Proj tool, or they could use it through their own RMT if this RMT has an implemented access to the catalogue by using the PABRE-WS web services.

PABRE-Proj goal is to facilitate the definition of the requirements for a particular IT project. During the requirements elicitation and specification, the tool helps requirements analysts in the browsing of the SRP catalogue and the application of SRPs. In the case of requirements not covered by any SRP, the tool allows to create new requirements related to an SRP or even with no relationship to any SRP. Once the elicitation finishes, the tool allows the generation of the SRS document, as well as a report with the SRP usage and data related to the new requirements (that will be used in the catalogue update). PABRE-Proj has both a desktop and a web version, to facilitate the access to requirements analysts when meeting customers. Although the interface is a little bit different, the functionalities provided by both of them are the same.

PABRE-WS is an API of web services that provides access to the SRP catalogue. The idea is to allow existing RMTs in the market to access the catalogue and to use SRPs in their own way during requirements elicitation and specification.

Finally, RE experts may use PABRE-Man as the SRP management tool. Its goal is to facilitate the creation and update of the SRP catalogue. With this tool, RE experts can add new SRPs, analyse the SRP usage data provided by PABRE-Proj to update the catalogue, and maintain the classification schemas used to organize the SRPs. As a way to support the work of the RE experts that manage SRPs, the tool provides a thesaurus that proposes changes in terminology, which improves the wording and the consistency among SRPs. PABRE-Man has been developed as a desktop application.

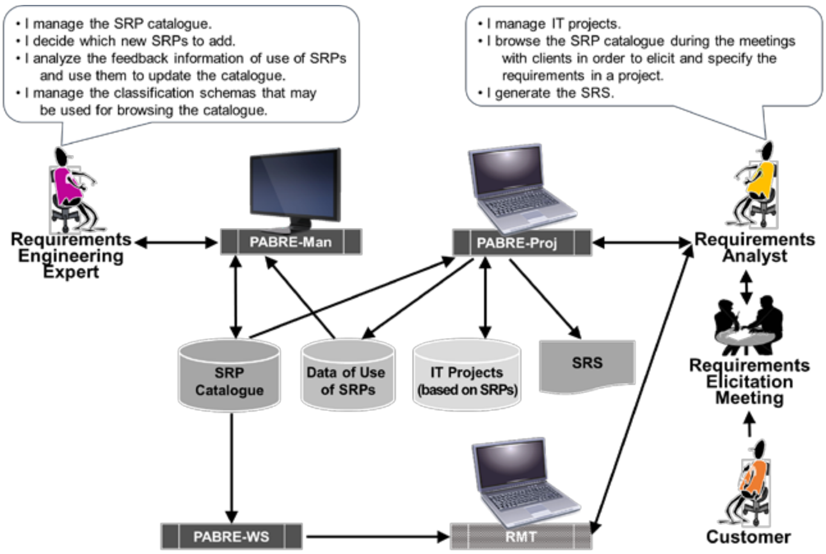


Figure 9.1 – The PABRE system

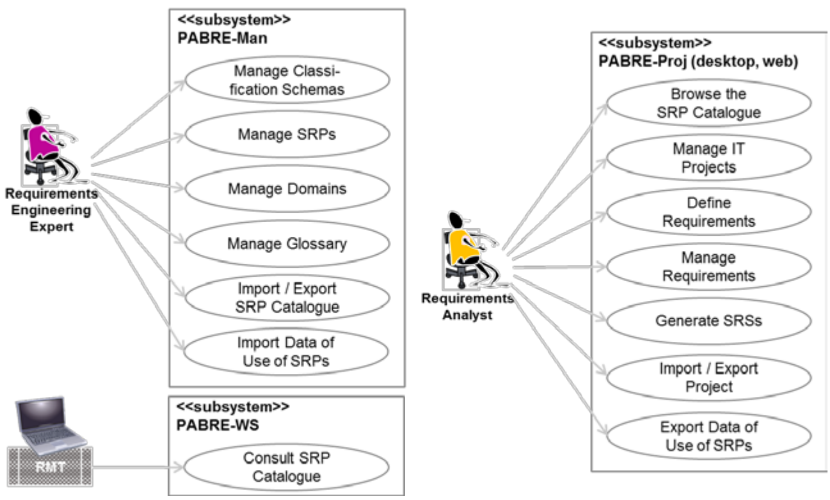


Figure 9.2 – Use case diagram of the PABRE system

Regarding the development, all technologies and libraries that have been used in the three subsystems are open source and have licenses that are flexible enough to allow the use of the systems in both open source and commercial projects. These licenses are: GNU-

GPL[23] (GNU General Public License), GNU-LGPL[24] (GNU Lesser General Public License), EPL[25] (Eclipse Public License) and APL[26] (Apache Public License). The architecture and technologies used in the three subsystems are explained below (Figure 9.3).

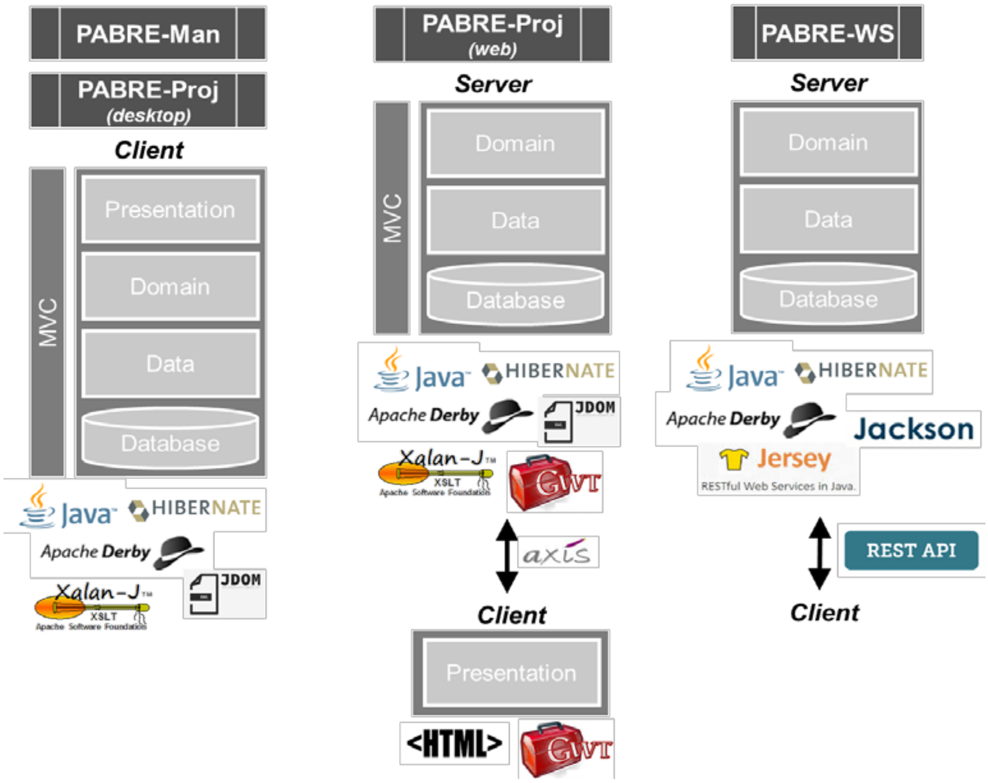


Figure 9.3 – PABRE subsystems architecture and technologies

PABRE-Man and PABRE-Proj (in its desktop version) are local systems based on a classical 3-layer architecture (Presentation, Domain and Data layers) and implement the design pattern Model-View-Controller (MVC). This architecture facilitates that the implementation can be carried out independently in each level (initially defining APIs or controllers that separate each level), and that changes in a level do not have to be propagated to the rest due to the logic separation that exists between levels.

Regarding technologies, both systems are developed using Java[27] (JDK version 6.24 Sun Microsystems), one of the more known object oriented programming languages, and

[23] http://www.gnu.org/licenses/gpl-3.0.en.html
[24] https://www.gnu.org/licenses/lgpl-3.0.en.html
[25] https://www.eclipse.org/legal/epl-v10.html
[26] http://www.apache.org/licenses/LICENSE-2.0

Hibernate[28] (Red Hat version 3.3.2), which is a framework for mapping the domain model to a relational database. For information persistence, Apache Derby[29] (version 10.5.3) is used as relational database system manager, as it allows an easy distribution of the systems when the databases are local, but also allows to be used easily for remote databases.

The most used Java libraries in these applications are JDOM[30] (version 1.1), to manipulate, access and collect Java objects from XML files, and Xalan-Java[31] (version 2.7 of Apache Project), an XSLT (eXtensible Stylesheet Language Transformation) processor to transform XML documents to HTML, text or other types of documents.

PABRE-Proj (in its web version) is also based on a classical 3-layer architecture implementing the MVC design pattern. In this case, as it is a client-server application, the client (in this case the browser) hosts the presentation layer, while the server hosts the domain and data layer, as well as the database, which in this case is remote.

Apart from Java, Hibernate and Apache Derby as in the previous systems, this web application also uses HTML, for defining the content of the web pages, and Google Web Toolkit[32] (GWT) (version 2.4) as a framework to create the JavaScript front-end code.

The most important Java libraries in this application, in addition to JDOM and Xalan-Java, is Apache Axis[33] (version 1.4), which implements the SOAP (Simple Object Access Protocol) protocol for exchanging structured information in distributed and decentralized environments, based on XML.

PABRE-WS is considered a client-server application, developed following a REST[34] (Representational State Transfer) architecture for distributed network applications. REST uses a communication protocol that is cacheable, stateless and of type client-server, which is usually the HTTP protocol, with the aim of making requests between systems in a simpler manner in comparison to other existing mechanisms such as CORBA or SOAP. As in most client-server applications, in REST architectures the client initiates a request to the server, and the server processes it and returns a response; the requests and responses

27 https://www.java.com/
28 http://hibernate.org/
29 http://db.apache.org/derby/
30 http://www.jdom.org/
31 https://xml.apache.org/xalan-j/
32 https://code.google.com/p/google-web-toolkit/
33 http://axis.apache.org/
34 http://www.ibm.com/developerworks/library/ws-restful/

are built as the representation of resources or objects. In the case of web services, there is no presentation layer, which would be hosted on the client, as it is not necessary, and the server hosts the domain and data layer, as well as the database, which is remote too.

Like the rest of the other PABRE systems, the implementation of PABRE-WS uses Java, Hibernate and Apache Derby. In this case, Jersey[35] (version 1.17) has also been incorporated, which is the reference implementation for JAX-RS Oracle (the Java API for RESTful Web Services), a programming language that provides support to create web services that follow the REST architecture.

Apart from JDOM, the Java library Jackson[36] (version 2.1.4) has also been used to convert POJO (Plain Old Java Object) and JSON (JavaScript Object Notation) objects.

The following sections explain the use of the PABRE system during the RE stage. Specifically, Section 9.2 presents how it is used during requirements elicitation and specification processes, and Section 9.3 explains how it is used for the creation and update of an SRP catalogue.

9.2 PABRE System for the Requirements Elicitation and Specification

This section shows how the PABRE system is used during the elicitation and specification of requirements for a specific IT project. For that purpose, the PABRE-Proj subsystem is only required (either in its desktop or web version), since it is the system focused on using the SRP catalogue to elicit and specify requirements. Through the example shown, the requirements analyst will be using the web version of the system, and it will be focused on the elicitation and specification of requirements related to the *Supplier Economic Information* NT-SRP (Table 7.6 in Chapter 7).

Before the requirements elicitation itself, the requirements analyst will come across with certain general project information, such as its domain and general overview, but also some parameters' values that can be fixed at the beginning of the project, since they are the same for the whole project, like the preferred currency, language or default user profiles (Figure 9.4). The PABRE-Proj screenshot to enter this first project information can be seen in Figure 9.5. The left side of the window contains the classification schema the requirements analyst selected for this

[35] https://jersey.java.net/
[36] https://github.com/FasterXML/jackson

project (in this case the one based on the ISO/IEC 9126 standard [137]), while the right side shows the part to edit the metadata of the project, with the name and description fields; although the project creation and modification dates are shown, they cannot be modified.

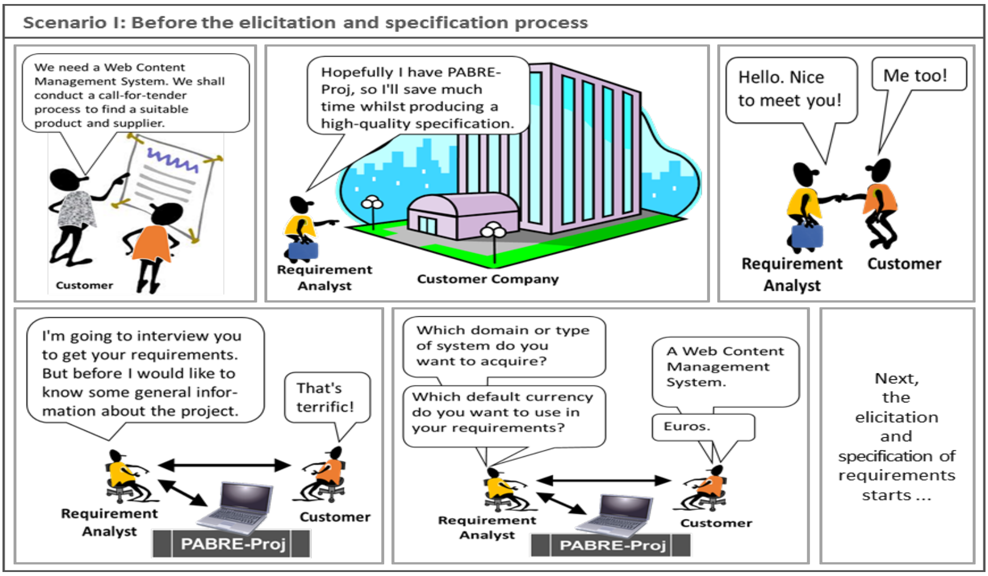


Figure 9.4 – Using the PABRE system before starting the elicitation and specification of requirements

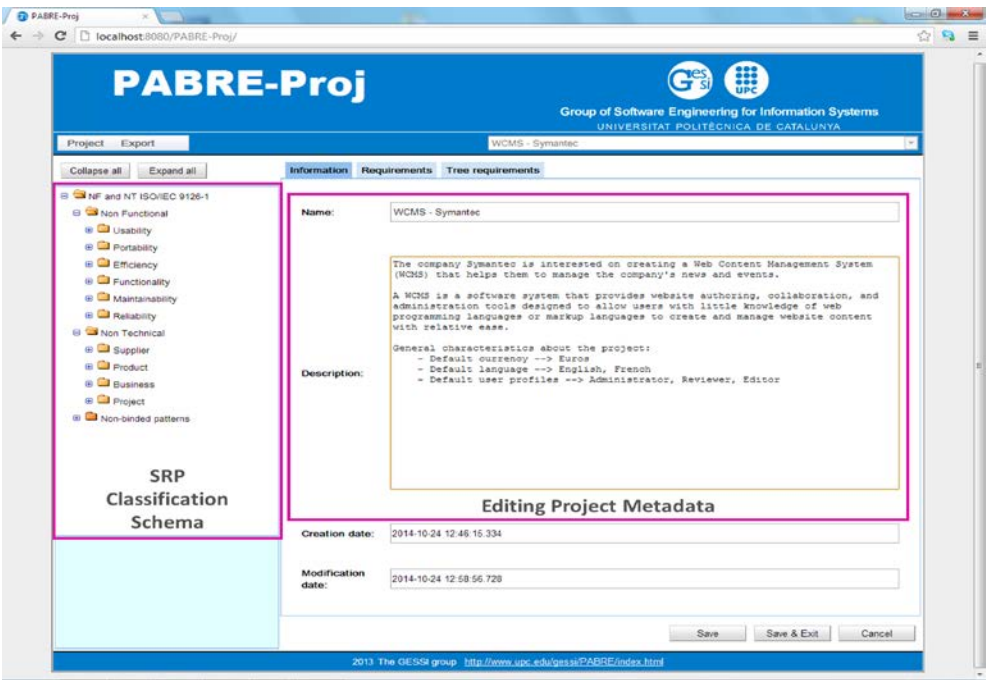


Figure 9.5 – Entering IT project information with PABRE-Proj

After entering the project metadata, the requirements elicitation and specification starts (Figure 9.6). Once an SRP has been selected (either by following the scope of the classification schema, arriving to it because an already used SRP is related to it or because the customer states a need and the requirements analyst looks for an SRP in the catalogue to fulfil it), the requirements analyst proposes it to the customer by asking if its goal helps to meet their needs. In case of the example, the selected SRP is the *Supplier Economic Information*. If the customer is interested in its goal, the requirements analyst explains the different clusters of the SRP. After the customer chooses one of the clusters, the requirements analyst enumerates its requirement templates and agrees with the customer which ones are the best ones for the project. The only missing part to state the requirements is to fill the parameters, taking into account the default values that have been fixed at the beginning of the project, which in case of the example is the currency. In order to create requirements that are the direct application of an SRP with PABRE-Proj, the requirements analyst has to choose first an SRP part in the tree showing the SRP catalogue (Figure 9.7) (in the example that would be the *Minimum Company Net Income* extended part of the *Economic Situation Prerequisites* cluster); and second, fill the parameters' values using the information provided by the customer (Figure 9.8). The three requirements stated for the *Supplier Economic Information* SRP in this scenario are shown in Figure 9.9.

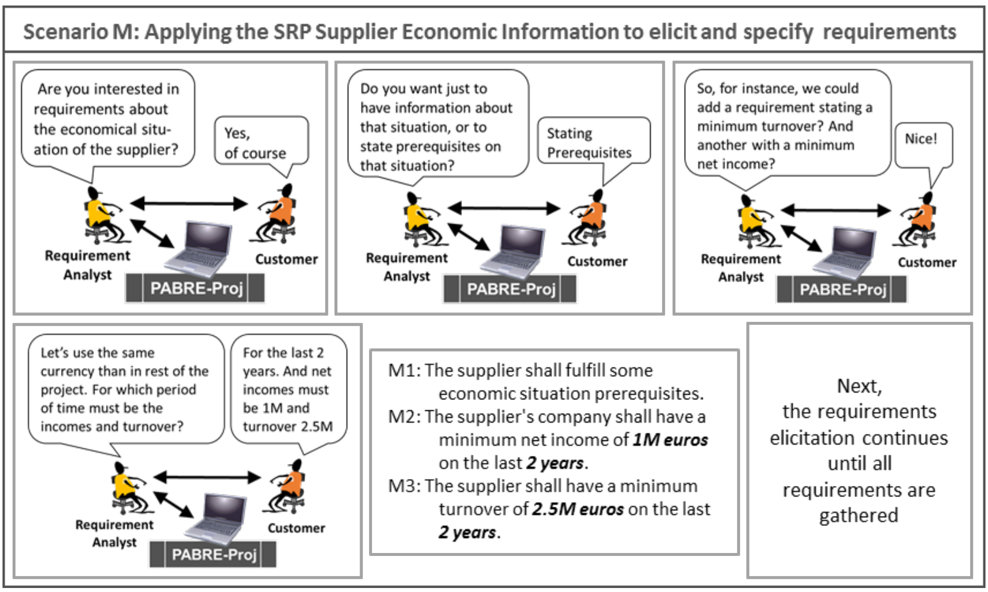


Figure 9.6 – Using the PABRE system for applying an SRP

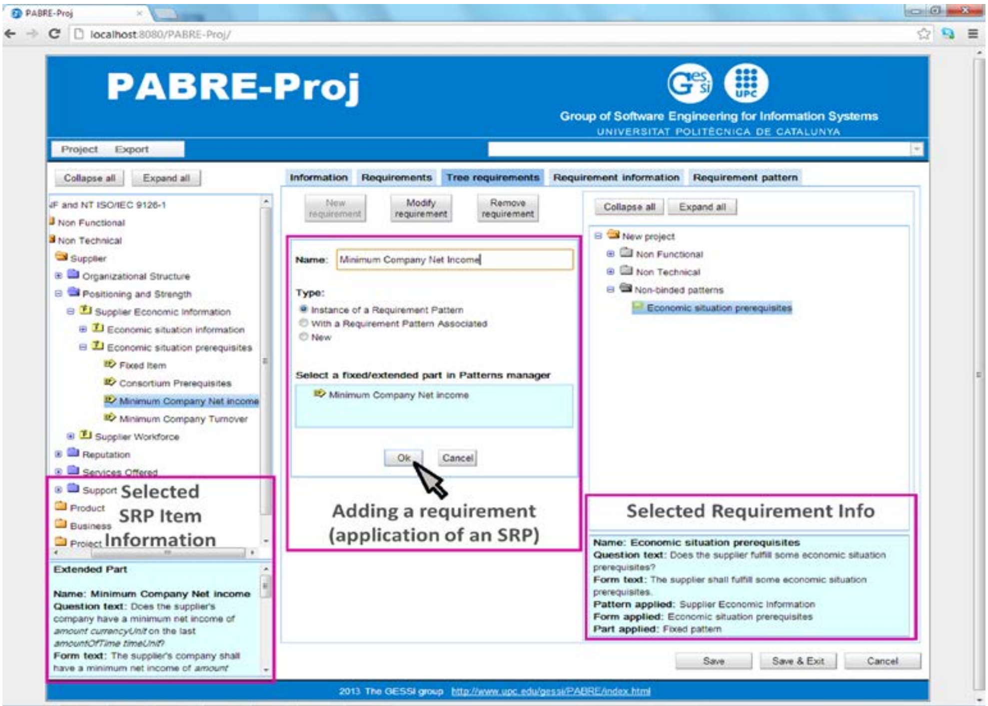


Figure 9.7 – Creating a requirement as an application of SRP in PABRE-Proj

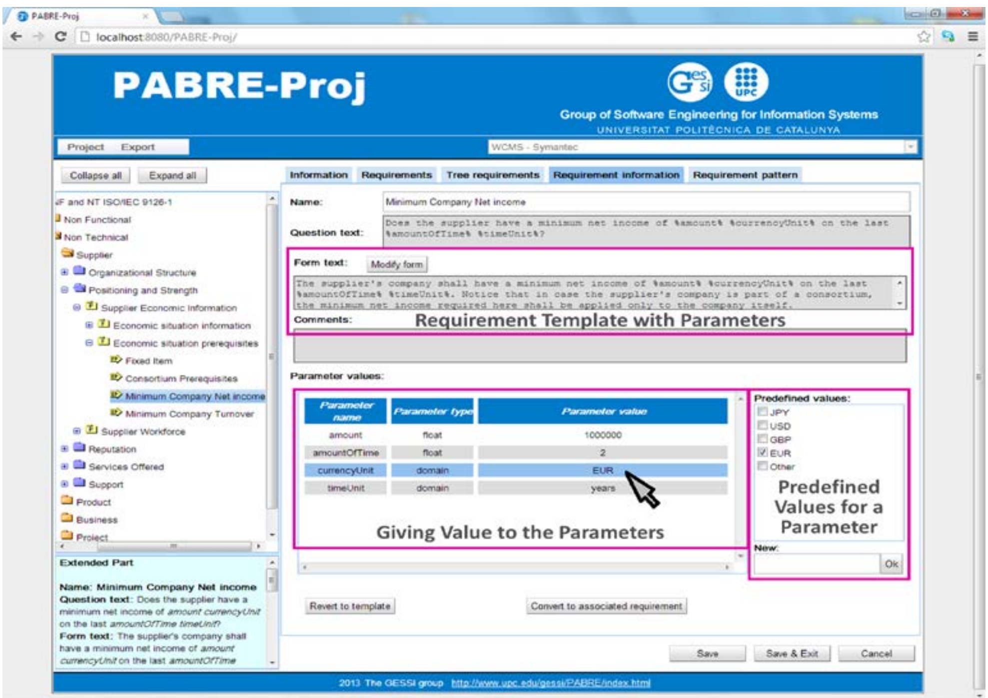


Figure 9.8 – Giving values to the parameters of an SRP application requirement in PABRE-Proj

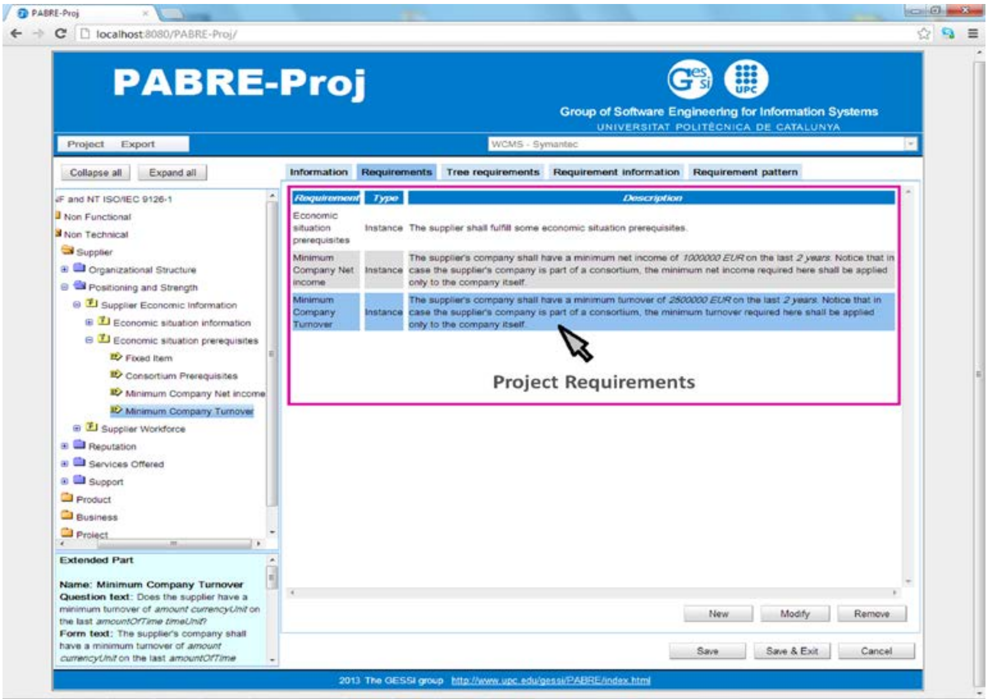


Figure 9.9 – Browsing the elicited and specified requirements in PABRE-Proj

However, while eliciting and specifying requirements, other situations may happen (Figure 9.10):

- The customer may not like the way the requirement templates are stated (scenario C).
- An SRP cluster may miss some requirements (parts) the customer needs (scenario B).
- An SRP may miss a cluster that address the goal as the customer wants (scenario A).
- The customer may be interested in a requirement that is not covered by any SRP (scenario N).

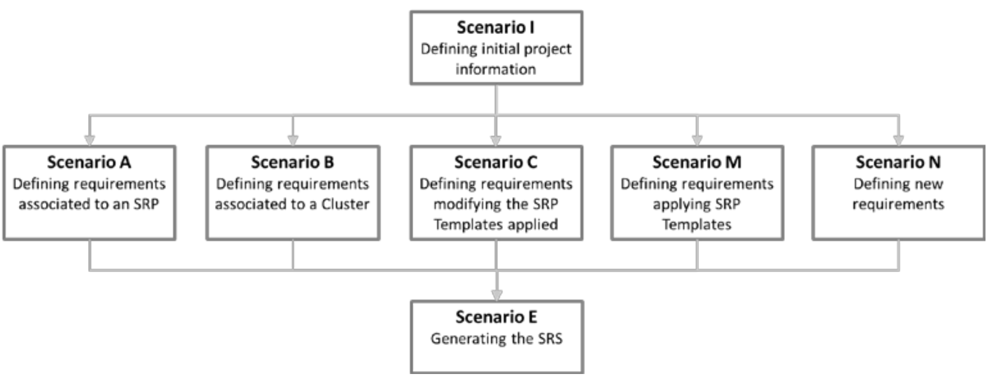


Figure 9.10 – Main scenarios while eliciting and specifying requirements with the PABRE system

All the previous situations can be addressed with the PABRE-Proj subsystem. For a matter of space, these scenarios are not described in this document, but the interaction with the PABRE-Proj tool in these scenarios (are well as in the scenarios already presented) can be seen in the following video: https://youtu.be/oZmUpfUS-KE.

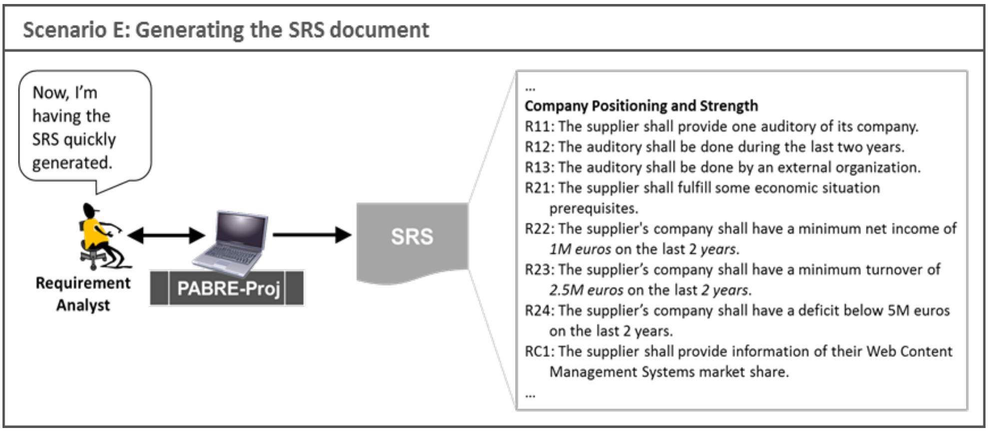
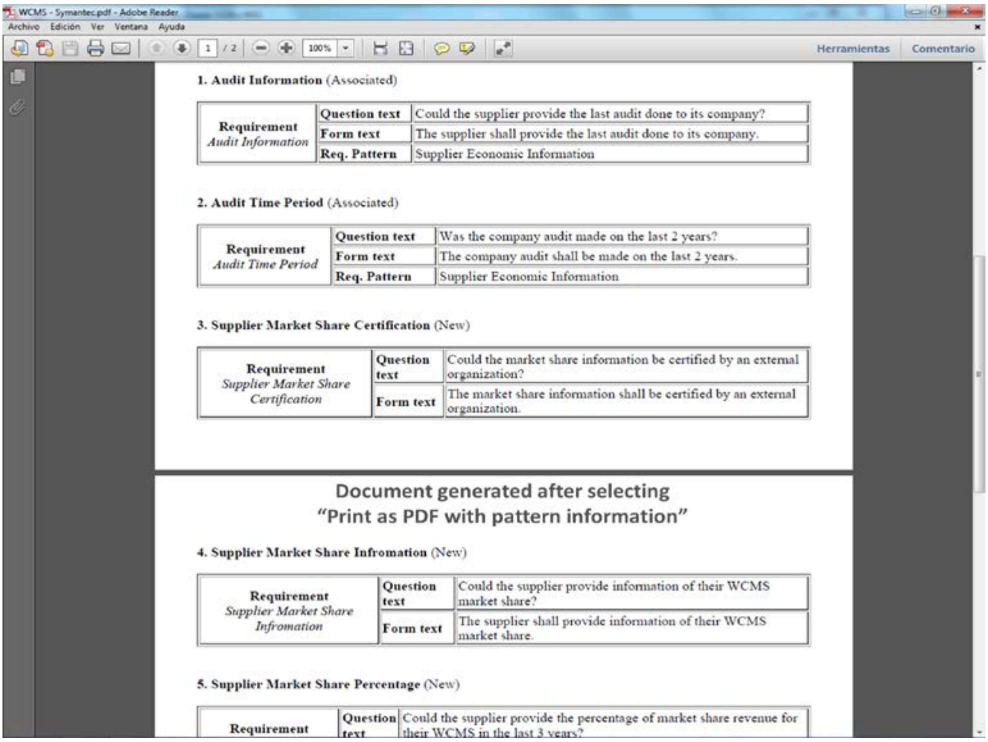


Figure 9.11 – Using the PABRE system for generating the SRS

Figure 9.12 – Project export functionalities in PABRE-Proj


WCMS - Symantec.pdf - Adobe Reader
Archivo Edición Ver Ventana Ayuda
Herramientas Comentario

1. Audit Information (Associated)

Requirement *Audit Information*	**Question text**	Could the supplier provide the last audit done to its company?
	Form text	The supplier shall provide the last audit done to its company.
	Req. Pattern	Supplier Economic Information

2. Audit Time Period (Associated)

Requirement *Audit Time Period*	**Question text**	Was the company audit made on the last 2 years?
	Form text	The company audit shall be made on the last 2 years.
	Req. Pattern	Supplier Economic Information

3. Supplier Market Share Certification (New)

Requirement *Supplier Market Share Certification*	**Question text**	Could the market share information be certified by an external organization?
	Form text	The market share information shall be certified by an external organization.

Document generated after selecting
"Print as PDF with pattern information"

4. Supplier Market Share Infromation (New)

Requirement *Supplier Market Share Infromation*	**Question text**	Could the supplier provide information of their WCMS market share?
	Form text	The supplier shall provide information of their WCMS market share.

5. Supplier Market Share Percentage (New)

Requirement	**Question text**	Could the supplier provide the percentage of market share revenue for their WCMS in the last 3 years?

Figure 9.13 – PDF generated with PABRE-Proj corresponding to the SRS

Apart from creating requirements, the requirements analyst can also modify or delete both projects and requirements (of any type).

Once the elicitation and specification finish, PABRE-Proj facilitates the generation of the SRS document as well as the exportation of requirements to some RMTs (such as OPAL [205]) (Figure 9.11). In the left-upper corner of Figure 9.12 it can be seen all the export formats that can be applied to a project in PABRE-Proj. If *Print as PDF with pattern information* is selected, a pdf document containing the SRS with all the requirements (applied or new) as well as all the SRPs that have been applied in the project is created (Figure 9.13).

9.3 PABRE System for the Creation and Update of an SRP Catalogue

In order to manage an SRP catalogue (both its creation and update), the PABRE-Man subsystem is needed. The PABRE-Proj subsystem is necessary just for exporting the report of use of the SRP catalogue for a specific IT project. This section explains how to use the PABRE system from the SRP management perspective. To do so, two scenarios are introduced: first, how it is possible to create an SRP, and second, how to update the SRP catalogue. For a matter of space, more scenarios of use of PABRE-Man are not described in this document, but a complete demo of the subsystem can be seen in the following video: https://youtu.be/zcTNb6MYaZ4.

The scenario for creating an SRP is shown in Figure 9.14. Before creating an SRP, the RE expert can open the information that could be useful to have at hand while creating that SRP. Specifically, PABRE-Man allows to consult three types of information at the same time: classification schemas (which include the SRPs too), a glossary of terms, and domains for the parameters (from left to right in Figure 9.15). The three panels showed in the figure not allow only to browse this information, but also give access to their management (creating, updating or deleting entities of each type).

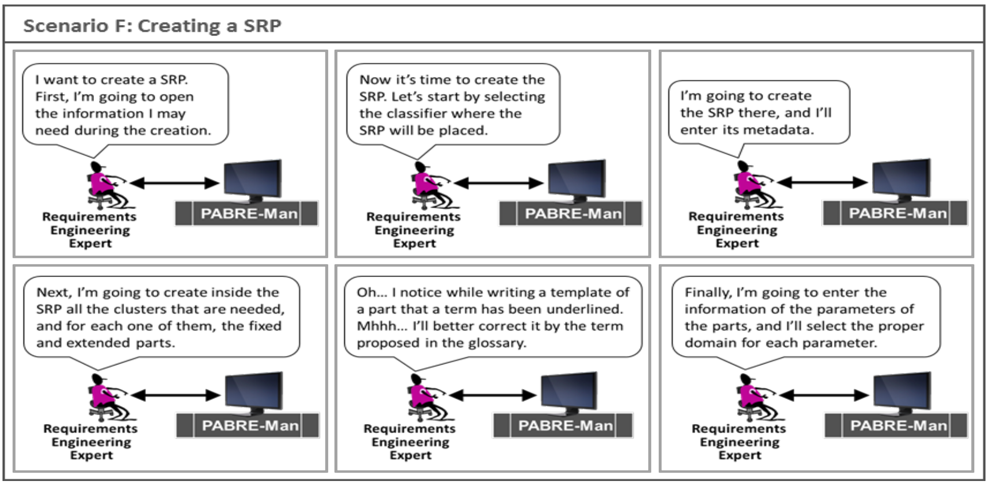


Figure 9.14 – Using the PABRE system to create an SRP

In order to create an SRP, the RE expert has to select a classifier where the SRP will be placed, and then select the option to create a new SRP. From the SRP creation window, the RE expert can enter the SRP metadata, create as much clusters as needed in the SRP, and for each cluster, enter the template for the fixed part and its extended parts, as well as its behaviour. Figure 9.16 shows the window to create an SRP (which is the same for editing it). In that case, the RE expert is creating the *Supplier Economic Information* SRP (Table 7.6 in Chapter 7), and specifically s/he is editing the *Minimum Company Turnover* extended part of the *Economic Situation Prerequisites* cluster. It is possible that while writing a template, a word is underlined (as noticed in the figure). This means that the underlined word has a preferred synonym defined in the glossary. By right-clicking over the word, it is possible to see the preferred synonym proposed by the glossary and substituting it. As can be seen in the same figure, the windows opened in the previous step are permanently available, so if the RE expert wants to consult another SRP, or even the glossary or a domain, s/he can do it.

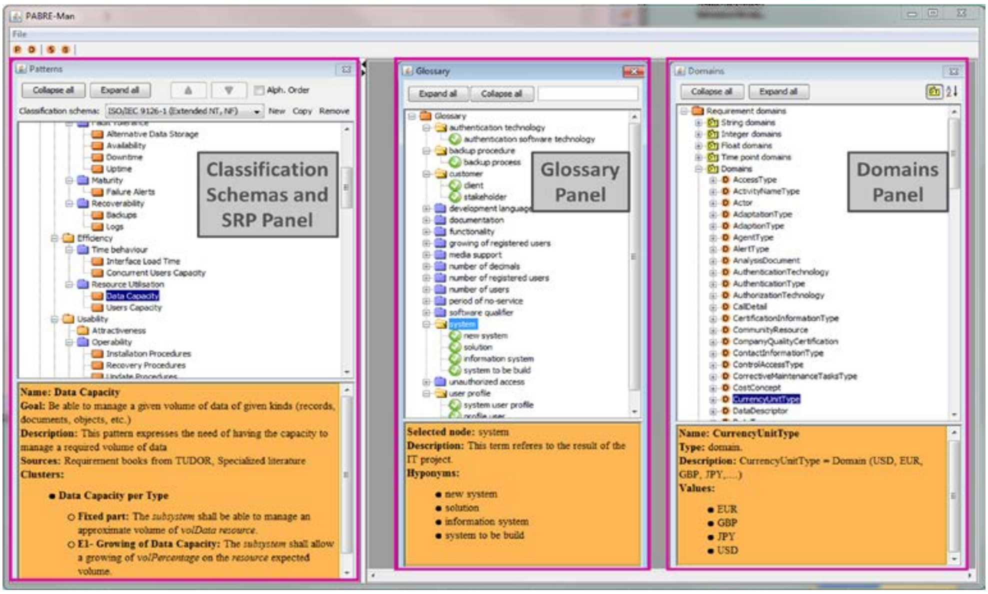


Figure 9.15 – Useful information while creating an SRP in PABRE-Man

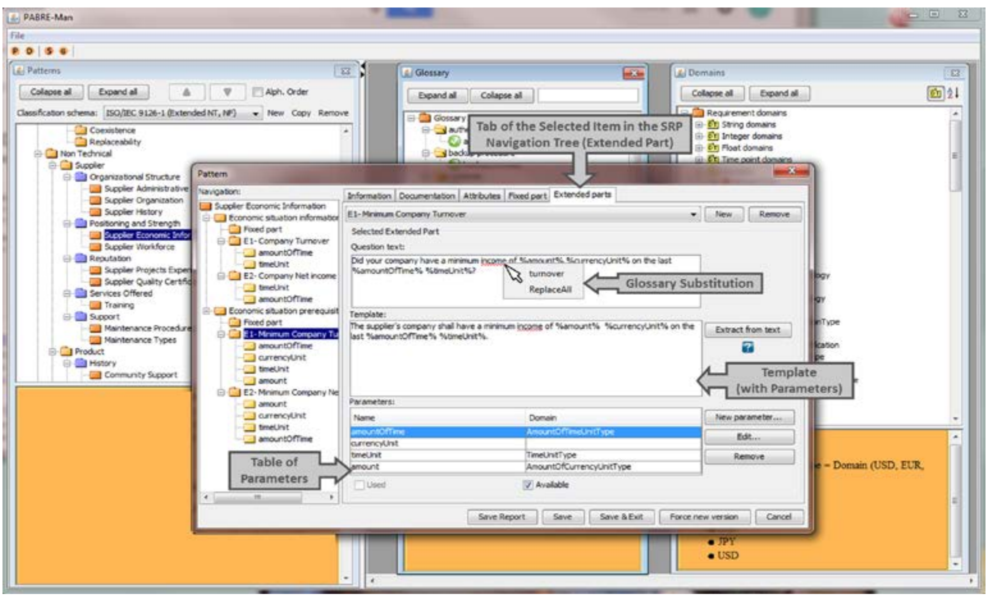


Figure 9.16 – Editing an SRP template with PABRE-Man

The only missing part to finish the creation of the SRP is entering the information about each one of the parameters that appear on the templates of the parts of the clusters, as well as selecting the proper domain for each one of them. If when looking for a domain for assigning it to a parameter, any of the available domains define the values that the RE expert is looking for, it is also possible to define a new one. Figure 9.17 shows the SRP creation window when a parameter is being edited, specifically the *currencyUnit* parameter of the *Minimum Company Turnover* extended part. The information necessary for a parameter includes its name, invariant and description (tab *Information* in the figure). The domain of the parameter is assigned in the *Attributes* tab, which is the one showed in the same figure.

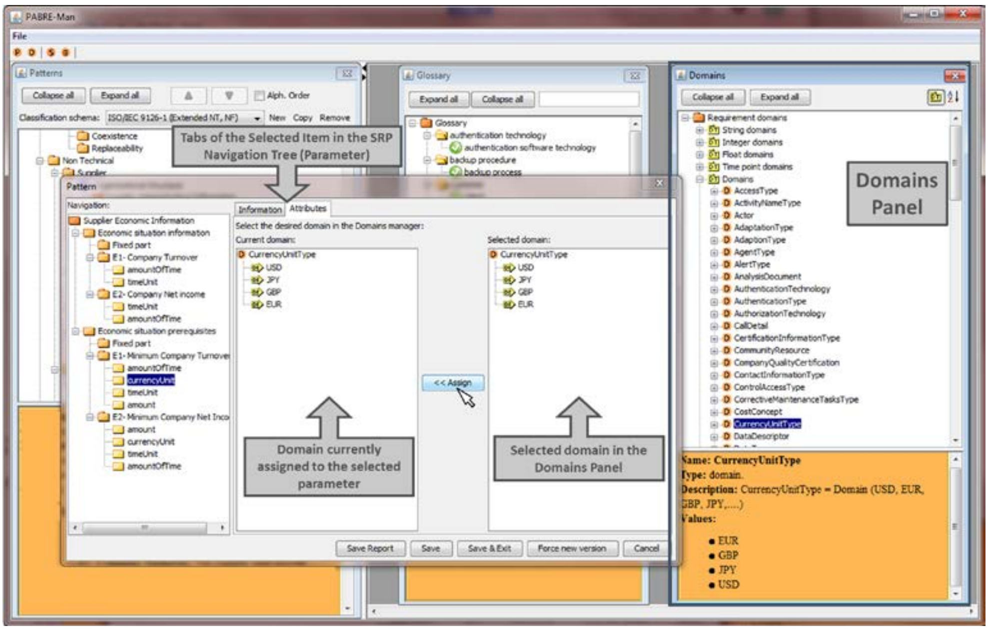
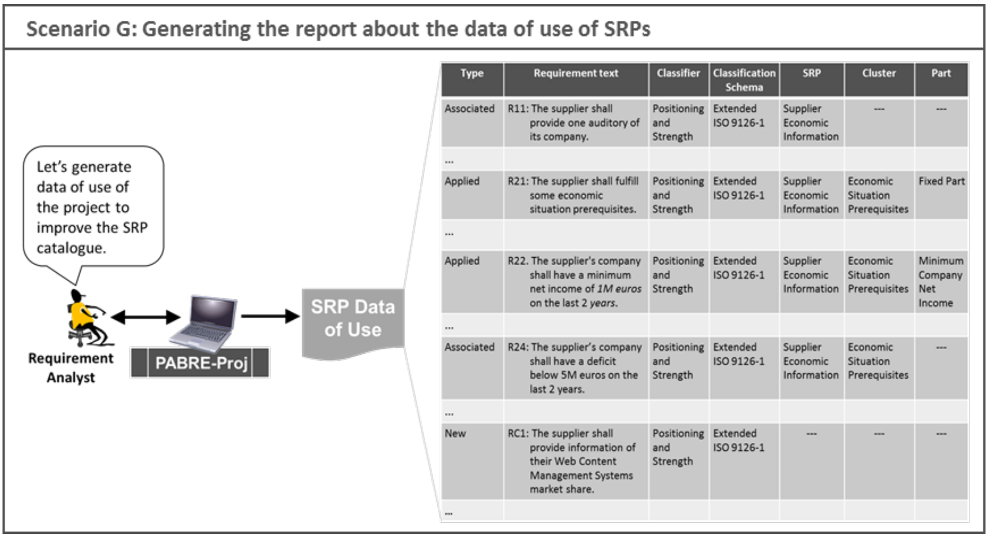


Figure 9.17 – Editing a parameter with PABRE-Man

In order to update the SRP Catalogue, PABRE-Proj allows exporting data about the SRP usage, as well as information about the requirements that have been created from scratch (which sometimes are associated to an SRP or one of its cluster) (Figure 9.18). Figure 9.19 shows an example of a scenario that takes place when updating the catalogue. Once the RE expert has imported to PABRE-Man the information about various projects, the update of the catalogue can start.

- First, the RE expert will look with the help of PABRE-Man if it is necessary to create new SRPs coming from the new requirements that are not related to any SRP.



Type	Requirement text	Classifier	Classification Schema	SRP	Cluster	Part
Associated	R11: The supplier shall provide one auditory of its company.	Positioning and Strength	Extended ISO 9126-1	Supplier Economic Information	---	---
...						
Applied	R21: The supplier shall fulfill some economic situation prerequisites.	Positioning and Strength	Extended ISO 9126-1	Supplier Economic Information	Economic Situation Prerequisites	Fixed Part
...						
Applied	R22. The supplier's company shall have a minimum net income of *1M euros* on the last *2 years*.	Positioning and Strength	Extended ISO 9126-1	Supplier Economic Information	Economic Situation Prerequisites	Minimum Company Net Income
...						
Associated	R24: The supplier's company shall have a deficit below 5M euros on the last 2 years.	Positioning and Strength	Extended ISO 9126-1	Supplier Economic Information	Economic Situation Prerequisites	---
...						
New	RC1: The supplier shall provide information of their Web Content Management Systems market share.	Positioning and Strength	Extended ISO 9126-1	---	---	---
...						

Figure 9.18 – Using the PABRE system to generate the report about the SRP data of use

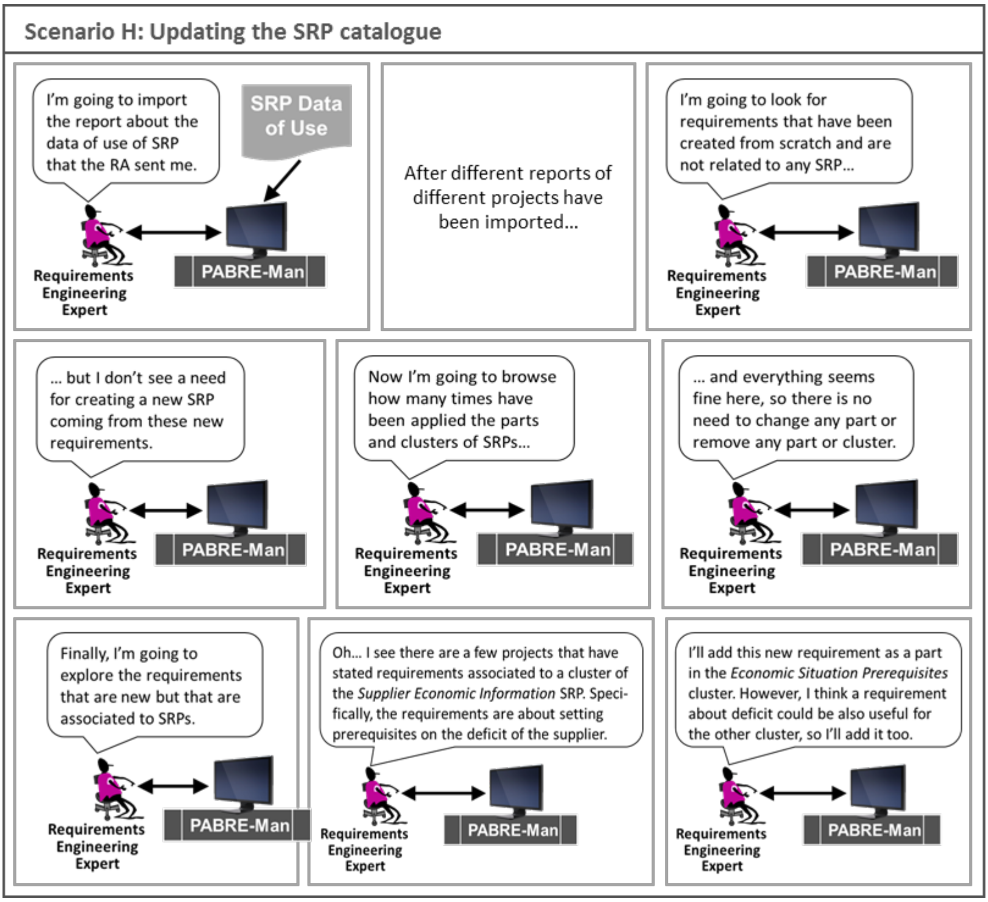


Figure 9.19 – Using the PABRE system to update the SRP catalogue

- Second, the RE expert will look for the information of use of SRPs, to see how many times a part, a cluster or a domain value of a parameter in an SRP has been used, as well as the modifications needed in the text of the templates when applying them in the projects or the new values needed in the parameters. On the one hand, a low usage of a part, cluster or parameter's value may imply its removal. On the other hand, modifications on the text on the templates when applying them may end in changing the templates in SRPs, and new parameter's values may suppose modifications on the domain assigned to that parameter.
- Third, the RE expert will explore if there are new requirements associated to an SRP or one of its clusters. In the first case, if a few projects need requirements to solve the goal of an SRP in a different way that the ones already proposed in the SRP, this may suppose the creation of a new cluster inside the SRP. In the second case, if there are a few projects that need the same type of requirement on the same cluster, this may imply the creation of a new extended part inside the cluster.

In the case of the example, the RE expert does not find appropriate to do any changes on the catalogue as a result of the first and second study. However, as a consequence of the third study, the RE expert notices that there are a bunch of requirements related to the *Economic Situation Prerequisites* cluster of the *Supplier Economic Information* SRP. Specifically, these requirements are stating conditions on the deficit the supplier has to have. Figure 9.20 shows the PABRE-Man window that allows browsing the data of use of SRPs and the new requirements created in the projects (associated or not to an SRP). In the case of the figure, only the data about the *Supplier Economic Information* SRP is being showed (the requirements either associated to the SRP or that are an application of one of its parts), since a filter to see only the information about this SRP is being used.

As consequence of the associated requirements to the *Economic Situation Prerequisites* cluster, the RE expert thinks a new extender part is needed on the cluster. Therefore, s/he modifies the SRP and incorporates this new part. In addition, the RE expert also believes that a part related to the deficit could be useful in the other cluster of the same SRP (the *Economic Situation Information* cluster, which just ask to the supplier to provide information about its economic situation). Subsequently, a new extended part is added to this cluster too.

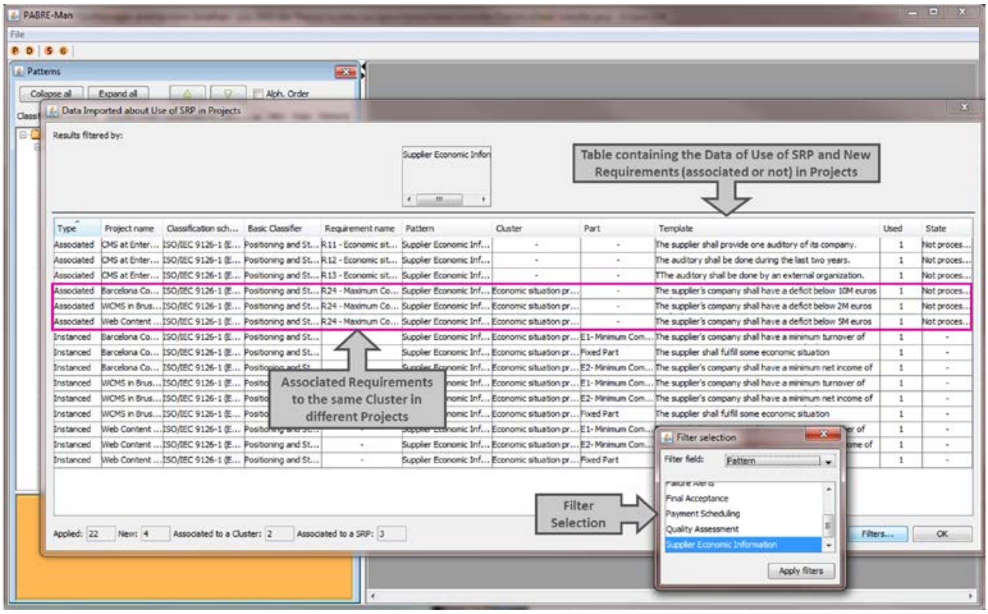


Figure 9.20 – Browsing the requirements associated to an SRP and its data of use with PABRE-Man

Part II

Validation of the PABRE Framework

CHAPTER 10 Validation of the SRP Metamodel

10.1 Introduction

In order to check the appropriateness of the SRP metamodel over all types of requirements (i.e. functional, non-functional and non-technical), the metamodel was validated with different sets of requirements.

With that purpose, the same analysis done to derive the SRP metamodel (Chapter 4) was carried out over a set of NTRs and over a set of FRs to validate the metamodel with respect to these types. The validation for NFRs was not necessary, since they were already used in Chapter 4 to construct the metamodel, and therefore it could be assured that the metamodel was valid for this kind of requirements.

Specifically, the same observations for each statement included in Chapter 4 were studied in NTRs and FRs, with the aim of seeing if the consequences over the metamodel for each statement were still applicable for NTRs and FRs, concluding for each statement if the consequences were supported or not by the observations.

The sets of requirements used in this analysis corresponded to the non-technical and functional parts of the same SRSs used in Chapter 4 (which were also the same as the ones used to construct the NT-SRP and F-SRP catalogue of the PABRE framework presented in Section 7.3 and 7.4, respectively).

In addition, the quality of the SRP metamodel was also analysed. The quality of a metamodel is related to the trade-off between completeness and complexity: it should be expressive enough but at the same time not overloaded with too many concepts. Because of that, the quality of the SRP metamodel was analysed with respect to two frameworks for metamodel quality.

The rest of the chapter presents both analyses. Concretely, Section 10.2 explores the results of the observations done over NTRs and FRs, showing the figures and providing a discussion over them. Section 10.3 studies the quality of the SRP metamodel. The conclusions of the analyses are presented in Section 10.4.

10.2 Analysis with Respect to NTRs and FRs

The results of the validation are presented in Table 10.1, and for each statement a conclusion is included regarding if the observations over NTRs and FRs support the derived SRP metamodel. In the following, a discussion is made over the most relevant observations.

The number of NTRs and FRs (Statement 1, Observation B) was higher than the number of NFRs. In the case of FRs, it was the highest one, which is not surprising taking into account that those type of requirements specify what a system should do, therefore supposing the differentiating factors among systems. For the NTRs, the number was higher because the SRSs corresponded to call-for-tender projects, and in this kind of project NTRs have a greater importance.

Regarding the proportion of reusable concepts (Statement 1, Observation D and E) and non-reusable concepts (Statement 1, Observation F and G), most of the non-technical concepts were reusable (38 concepts, 93%) due to the fact that these type of requirements usually do not depend neither on the type of system nor on its domain. Although the same could be thought about NFRs, in that case there were much more non-reusable concepts appearing in just one SRS (13 concepts, 28%) if that number is compared to the same ratio in NTRs and FRs. This fact downsized the ratio of reusable concepts (28 concepts, 61%). As for FRs, the percentage of reusable concepts was in the middle of both cases (44 concepts, 82%).

One aspect surprising about FRs was the large number of reusable requirements (Statement 1, Observation H). If it is compared to the ratio of reusable NFRs and NTRs (78% and 86%, respectively), the percentage of reusable FRs was quite similar (78%). One could have expected the number of reusable FRs to be smaller since, at the end, this type of requirement differentiates one system from another. However, that was not the case in the systems of the analysed SRSs. The reason behind that is probably that as all the SRSs came from the same organization (i.e. LIST) and they corresponded to the same domain (i.e. CMSs), they were quite similar one from another. It is worth remarking, although, that in the 6 SRSs there were a total of 303 requirements that were not possible to reuse since they were too project specific, which corresponded to over 50 different requirements related to different functionalities in the analysed systems.

Table 10.1 – Results of observations made on NTRs and FRs

Statement		Observation	NFRs	NTRs	FRs	Conclusion
1	A	# SRSs	6	6	6	OK
	B	# requirements	533	785	1401	
	C	# concepts	46	41	54	
	D	# reusable concepts appearing in more than one SRS	26 (57%)	35 (86%)	42 (78%)	
	E	# reusable concepts appearing in one SRS	2 (4%)	3 (7%)	2 (4%)	
	F	# non-reusable concepts appearing in more than one SRS	5 (11%)	2 (5%)	9 (17%)	
	G	# non-reusable concepts appearing in one SRS	13 (28%)	1 (2%)	1 (2%)	
	H	# reusable requirements	417 (78%)	672 (86%)	1098 (78%)	
2	A	# requirements restricting entities	417 (100%)	323 (41%)	1098 (100%)	The relationship between *Entity Type* and *SRP* should be *-* (and not 1..* - *)
	B	# entity types	7	7	1	
3	A	# clusters	192	173	127	The relationship between *Entity Type* and *SRP* should be *-* (and not 1..* - *)
	B	# clusters restricting concepts	192 (100%)	173 (100%)	127 (100%)	
	C	# clusters restricting entities	192 (100%)	109 (63%)	127 (100%)	
	D	# minimum of requirements per cluster	1	1	1	
	E	# maximum of requirements per cluster	9	13	32	
	F	# clusters with one requirement	89 (46%)	34 (20%)	12 (9%)	
	G	# clusters with more than one requirement	103 (54%)	139 (80%)	115 (91%)	
4	A	# reusable concepts with different requirements clusters	6 (21%)	10 (26%)	23 (52%)	OK
	B	# reusable concepts with one requirements cluster	22 (79%)	28 (74%)	21 (48%)	
5	A	# requirements clusters with one requirement characterizing it and zero or more refining requirements	192 (100%)	173 (100%)	127 (100%)	OK
	B	# requirements clusters where the part that characterizes it was implicit	49 (26%)	41 (24%)	48 (38%)	
	C	# requirements clusters with some refining requirement	103 (54%)	139 (80%)	115 (91%)	
6	A	# requirements containing specific information about the context of the project	371 (89%)	336 (50%)	866 (79%)	OK
7	A	# requirements clusters with more than one refining requirement	54 (28%)	98 (57%)	98 (77%)	OK
	B	# requirements clusters where the same type of refining requirement appeared more than once	19 (10%)	25 (14%)	65 (51%)	

	C	# requirements clusters where the existence of a refining requirement did not make sense if another refining requirement was not present in the cluster	3 (2%)	3 (2%)	68 (54%)	
	D	existence of refining requirements in clusters of different SRSs that could not appear together in a requirements cluster	True	True	True	
8	A	# relationships between cluster - cluster, type: conflicts	3	-	-	OK
	B	# relationships between cluster - cluster, type: relates-to	55	174	147	
	C	# relationships between cluster - cluster, type: require	15	12	45	
	D	# relationships between cluster - cluster, type: supports	304	-	911	
	E	# relationships between requirement - requirement, type: relates-to	3	-	-	
	F	# relationships between project info - project info, type: supports	49	89	480	
	G	# relationships between project info - project info, type: disjoint	8	-	-	
	H	# relationships between project info - project info, type: rule	10	-	-	
	I	# relationships between project info - project info, type: subset	40	6	64	
	J	# relationships between project info - cluster, type: requires	4	80	-	
	K	# relationships between requirement - cluster, type: requires	-	10	23	
9	A	requirements could be classified in different ways	True (e.g. SRSs, GESSI, [4])	True (e.g. SRSs, [8])	True (e.g. SRSs, [177])	OK
10	A	hierarchical classification of requirements	True	True	True	OK
	B	# minimum of classifiers	5	5	3	
	C	# maximum of classifiers	9	10	11	
11	A	# classifiers in SRSs with one concept in the same SRS	4 (8%)	7 (16%)	1 (2%)	OK
	B	# classifiers in SRSs with different concepts in the same SRS	44 (92%)	37 (84%)	45 (98%)	
12	A	# requirements that could be classified in different classifiers	20 (5%)	111 (17%)	77 (7%)	OK

Analysing the number of requirements restricting entities (Statement 2, Observation A) an important difference was found in the case of NTRs. Some of these requirements did not restrict any entity (349 requirements, 52%) in comparison with NFRs and FRs, where all the requirements restricted one. The reason behind that is that some NTRs did not restrict any aspect of the system, like for instance the requirement of SRS-1 *We want to know the distribution of the workforce your company trades between the development and maintenance teams*. This caused that not all clusters of NTRs restricted an entity (Statement 3, Observation C), implying a change in the metamodel. The multiplicity of the relationships *constraints* and */has_as_entity_types* in the direction *SRP* and *SRP Cluster*, respectively, to *Entity Type* should be *0..** instead of *1..** (*1..** is what was deducted from the observations over NFRs in the Statement 3 of Subsection 4.2.1).

Another fact worth remarking in relationship with *Entity Types* is that FRs were only restricting one of them (Statement 2, Observation B), being the entity type the *Subsystem*. Also related to FRs, it is curious to see that there were only 127 clusters (Statement 3, Observation A) for 1098 requirements, while for NFRs and NTRs there were 192 and 173 clusters for 417 and 172 requirements, respectively. The reason behind that is that FRs are much more related among them and for the same concept (i.e. *Content Version Management*) there were more requirements that referred to it. This was corroborated by the fact that the maximum number of requirements per cluster (Statement 3, Observation E) was almost three times the maximum for NFRs and NTRs, and that for FRs the rate of clusters with just one requirement (Statement 3, Observation F) was much smaller than for the other types of requirements.

Looking at the number of reusable concepts with different requirements clusters (Statement 4, Observation A), the figure was almost the double in the case of FRs. This could be explained by the fact that functionalities can be achieved in more versatile ways, thus giving place to different clusters in different SRSs for the same concept.

Although for the three types of requirements all requirements clusters had one requirement that characterized it (Statement 4, Observation A), in the case of FRs that requirement was implicit in 38% of the cases (in comparison to 24% and 26% in NFRs and NTRs). This implies that FRs incorporated more implicit knowledge than the other two types. For instance, SRS-1 contained the requirement *A user should be able to edit contents from the publication space*. This implied that the system should incorporate an editor, and this editor probably should have some specific functions (it is not the same an editor with simple editing functions than an editor with advanced editing functions). However, SRS-1 did not include any requirement related to an editor, which was implicit knowledge in the SRSs.

As for the number of requirements containing specific information about the context of the project (Statement 6, Observation A), the number was relatively lower for NTRs (50% of the requirements) in comparison to NFRs and FRs (89% and 79%, respectively). As explained before, this probably was due to the fact that some NTRs did not restrict any aspect of the system, therefore not containing specific information for the project.

The number of clusters with more than one refining requirement (Statement 7, Observation A) was bigger in the case of NTRs and FRs. This difference could be explained by the fact that NTRs and FRs contained on average more requirements per cluster than NFRs (4 and 9 requirements, respectively, in contrast to 2). As a consequence, the number of clusters where the same refining requirement appeared more than once was also larger for NTRs and FRs (Statement 7, Observation B).

It is also worth remarking the larger number of functional requirements clusters where the existence of a refining requirement did not make sense if another refining requirement was not present in the cluster (Statement 7, Observation C). This could be, of course, related to the fact that in the case of FRs there were much more clusters with more than one refining requirement (otherwise, the situation could not appear). However, it was believed that the reason was more related to FRs being more cohesive among them (e.g. it does not make sense to ask a system to be able to delete a banner when its duration expires if the system is not able to establish a duration for banners).

Regarding the relationships among elements of different granularity studied in the observations of Statement 8 (i.e. clusters, requirements, and project specific information), the analysis over NTRs and FRs showed that not all the granularities and types of relationships appeared in all type of requirements (see, for instance, Observation E and J). In fact, while analysing NTRs and FRs, a new relationship with a different granularity arose (the one among requirement and cluster, Observation K). This corroborated the decision to make the part of the metamodel related to relationships as general as possible in Statement 8 of Subsection 4.3.1. It is also worth highlighting here that FRs presented a large number of relationships of type supports in two granularities: the one among requirement and requirement, and the one among project specific information and project specific information. This was explained not for having only a large number of FRs in the SRSs, but also because FRs (and the functionalities specified on them) shall be consistent and therefore support each other (i.e. if a requirement A specifies that a functionality X shall be accessible from a specific part of the interface, some requirement B must specify the functionality X, and therefore requirements A and B must be consistent and support each other).

NTRs and FRs could also be classified in different ways (Statement 9, Observation A). Specifically, it was found that apart from the classification used in the SRSs, NTRs could be classified using, for instance, the extension of the ISO/IEC 9126 with non-technical factors presented in Carvallo's et al. proposal [8], and FRs could be classified using, for instance, the quality model for CMSs proposed in Franch et al. [177].

All the observations in Statement 10 were true for NTRs and FRs. The only aspect to highlight here is that, in the case of FRs, there was an SRS with just 3 classifiers (Observation B), corresponding to *Integration with Tabellio Libellium*[37], *Word Processor Adaptation*, and *Site Management*. Taking into account that the functional part of the SRSs corresponded to a big part of the requirements, the classifiers contained a lot of requirements, which made the search for a specific requirement difficult in this specific SRS. The rest of SRSs contained at least 8 classifiers for FRs.

The number of classifiers that contained requirements related to just one concept in the same SRS was just one in the case of FRs (Statement 11, Observation A). Again, this was explained by the fact that the number of FRs was bigger in relation to NFRs and NTRs, that they were more cohesive among them, and that the number of relationships among them was bigger. All together increased the possibility of having requirements constraining different concepts under the same classifier in the same SRS, downsizing therefore the number studied in this observation.

The proportion of requirements that could be classified in different classifiers (Statement 12, Observation A) was similar in FRs and NFRs (5% and 7%, respectively). For NTRs, however, the ratio was a bit larger (17%). The main reason for that was that there were more non-technical concepts related to various aspects (usually, the project as a business and the supplier), and therefore the requirements related to these concepts could be placed under classifiers that were related to either the project or the supplier.

10.3 Analysis with Respect to Quality

The quality of the SRP metamodel was checked according to two different metamodel quality frameworks. The first one is proposed by López et al. [206]. In their work, quality features correspond basically to syntactic rules that metamodels should follow. These

[37] *Tabellio Libellium* is an application being used by the organization offering the call-for-tender project

features (see Table 10.2) are categorized in design flaws (properties signalling a faulty design, e.g. there should not be isolated classes), best practices (basic design quality guidelines, e.g. no class can be contained in two classes), naming conventions (related to the use of verbs, nouns or pascal/camel case, e.g. attributes should not be named after their feature class) and metrics (measurements of metamodel elements and their threshold value, e.g. the maximum number of attributes a class should reasonably define). The metamodel that this thesis presents complies with the 30 properties that compose the framework. Therefore, it is a metamodel of quality according to this framework.

Table 10.2 – Metamodel quality features from López et al. used for the SRP metamodel quality analysis

	Design metamodel quality features
1	An attribute is not repeated among all specific classes of a hierarchy.
2	There are no isolated classes (i.e. not involved in any association or hierarchy).
3	No abstract class is super to only one class (it nullifies the usefulness of the abstract class).
4	There are no composition cycles.
5	There are no irrelevant classes (i.e. abstract and subclass of a concrete class).
6	No binary association is composite in both member ends.
7	There are no overridden, inherited attributes.
8	Every feature has a maximum multiplicity greater than 0.
9	No class can be contained in two classes, when it is compulsorily in one of them.
10	No class contains one of its superclasses, with cardinality 1 in the composition end.
	***Best practices* metamodel quality features**
11	There are no redundant generalization paths.
12	There are no uninstantiable classes (i.e. abstract without concrete children).
13	There is a root class that contains all others (best practice in EMF).
14	No class can be contained in two classes (weaker version of property D09).
15	A concrete top class with subclasses is not involved in any association (the class should be probably abstract).
16	Two classes do not refer to each other with non-opposite references (they are likely opposite).
	***Naming conventions* metamodel quality features**
17	Attributes are not named after their feature class (e.g. an attribute paperID in class Paper).
18	Attributes are not potential associations. If the attribute name is equal to a class, it is likely that what the designer intends to model is an association.
19	Every binary association is named with a verb phrase.
20	Every class is named in pascal-case, with a singular-head noun phrase.
21	Element names are not too complex to process (i.e. too long).
22	Every feature is named in camel-case.
23	Every non-boolean attribute has a noun-phrase name.
24	Every boolean attribute has a verb-phrase (e.g. isUnique).
25	No class is named with a synonym to another class name.
	***Metrics* metamodel quality features**
26	No class is overloaded with attributes (10-max by default).
27	No class refers to too many others (5-max by default).
28	No class is referred from too many others (5-max by default).
29	No hierarchy is too deep (5-level max by default).
30	No class has too many direct children (10-max by default).

The second framework corresponds to the work of Bertoa et al. [207]. In this work, a rich set of quality attributes for metamodels, composed by more than 40 properties, is presented. As a consequence of the huge variety of properties proposed in this framework, only the most relevant ones were studied for the SRP metamodel. The relevant characteristics are the same chosen by Hinkel et al. [208] in their empirical study about the perception of the quality of metamodels (which also uses Bertoa's et al. framework as a reference): *complexity*, *self-descriptiveness*, *modularity*, *conciseness*, *completeness*, *correctness*, *changeability*, and *consistency* (Table 10.3 contains the definition of each one of these quality properties). In the following, a discussion about the compliance of the SRP metamodel according to the quality features of Table 10.3 is provided:

- **Complexity.** The metamodel was presented in different conferences, tutorials, workshops and in the interviews of the empirical study explained in Chapter 11, and during these presentations the questions and comments of the attendees reflected that they understood what SRPs were. This fact corroborates that the metamodel is not difficult to understand. In addition, it is possible to assert that there is a good trade-off between the complexity of the metamodel and its expressiveness, since the metamodel was derived from real SRSs.
- **Self-descriptiveness.** The self-descriptiveness of the metamodel is aided by having used UML for expressing it, which is a well-known language that eases, for instance, the visual description of the different object classes and the associations between them. In addition, the fact that the metamodel is accompanied by a glossary with all the concepts appearing on it (see Appendix III) makes the metamodel self-descriptive.
- **Modularity.** This feature is assured in the metamodel since, as presented in Chapter 4, it is comprised of three parts that can be understood separately: the core structure of SRPs, SRP relationships and SRP classification schemas.
- **Conciseness.** Provided that the metamodel was derived from the real SRSs, it is possible to affirm that SRPs are described to the point in the metamodel and there is no unnecessary information included in the metamodel.
- **Completeness.** As in the case of conciseness, it is possible to affirm that all statements are correct and relevant for SRPs because the metamodel was derived from the real SRSs.

Table 10.3 – Metamodel quality features from Bertoa's et al. framework used for the SRP metamodel quality analysis

Metamodel quality features	
Complexity	Measures the effort required to understand a metamodel; the extent of cognitive complexity of a metamodel measured by some index or indices.
Self-descriptiveness	The extent that a metamodel contains enough information for a reader to determine its objectives, assumptions, constraints, inputs, outputs, components, and status.
Modularity	The extent that the metamodel's parts are systematically structured and separated such that they can be understood in isolation.
Conciseness	The extent that the concept modelled in the metamodel is described to the point and not unnecessarily extensive.
Completeness	The metamodel contains all statements that are correct and relevant about the domain.
Correctness	The metamodel includes correct elements and correct relations between them and not violating rules and conventions. Thus it covers both syntactic correctness (right syntax or well-formedness) and semantic correctness (right meaning and relations relative to the knowledge about the domain).
Changeability	Of importance in a dynamic world is the changeability of metamodels when the domain or our understanding of it changes or the modelled concept must evolve because of changing requirements.
Consistency	The extent to which similar elements (classes, properties) of a metamodel are represented with the same structure, format, and precision.

- **Correctness.** It is possible to assure that the metamodel is syntactically correct since it was checked with respect to the metamodel quality framework of López et al. that, as stated before, mostly comprises syntactic rules for metamodels. In addition, it is certain that the metamodel incorporates the right meaning about SRP elements and relations relative to them, since the knowledge was extracted from real SRSs.
- **Changeability.** In the case of SRPs, the changeable part is the representation of the requirement concept. Right now, SRPs correspond to natural language requirement and they are encapsulated in the metamodel in the *template* attribute of the *Requirement Abstraction* class, its associated classes *Parameter* and *Domain*). If natural language requirements want to be changed, for instance, to data conceptual models or uses cases, the only parts to be modified in the metamodel would be these three concepts. Therefore, the change would be easy to make. Additionally, changes in the metamodel would be eased by the fact of having used UML to express the metamodel, as it is a well-known language to define metamodels.
- **Consistency.** All the classes of the metamodel, and their properties and relationships, are represented with the same structure and format, and the same granularity (i.e. precision).

10.4 Conclusions

The observations extracted from NTRs and FRs for the validation corroborate the observations made over NFRs, except in a point related to the *Entity Type* class associations. As stated before, it was necessary to modify the multiplicity of the relationships *constraints* and */has_as_entity_types* in the direction *SRP* and *SRP cluster*, respectively, to *Entity Type*. This modification was due to the observations over NTRs in Statement 2 (Observation A) and Statement 3 (Observation C), as some NTRs were actually not restricting any entity. The modification did not suppose a big change in the proposal, since it just implied making the metamodel less restrictive to allow that not all *SRPs* and *SRP Clusters* should be associated to *Entity Types*. As a consequence, *SRP Clust*ers will be applied in a project for zero entities, if there is no *Entity Type* associated to it, or for one or more specific entities, if there are *Entity Types* associated to the *SRP Cluster*. The implication of this observation has been already incorporated in the final version of the metamodel (the one showed in Chapter 4 Subsection 4.2.2).

Apart from that, the observations extracted from NTRs and FRs for the validation demonstrated that the metamodel presented in Chapter 4 is valid for NT-SRPs and F-SRPs, and therefore is valid for all types of requirements.

In addition, a quality analysis has been done over the SRP metamodel, showing that the metamodel proposed is a good metamodel for SRPs.

CHAPTER

11 Validation of the PABRE Framework

11.1 Introduction

In order to validate the acceptance of the overall PABRE framework, and to corroborate some of the results obtained in the survey-based empirical study presented in Section 2.3, another empirical study was carried out to investigate the perception of participants about the usefulness, benefits and drawbacks of the PABRE framework. With that purpose, semi-structured interviews of around two hours were conducted to 22 RE professionals of 11 Swedish organizations. The framework was presented individually to the participants during the interviews, where they were able to solve any doubt that might arise.

As stated in the related work of requirements reuse empirical studies (Subsection 2.3.1), only a few empirical studies addressing specifically requirements reuse by means of patterns were found, which correspond to the ones of Hoffmann et al. [55], Issa et al. [113], Karatas et al. [114], Pacheco et al. [115] and Toval et al. [116]. Hoffmann's et al. study is based on few interview's answers, and it is just exploring the general viability of RPs as a reuse approach (no specific technique was explained to the interviewees, just the general concept of RP). Toval's et al. approach is similar to the SRPs proposed in PABRE, but their case study is mainly focused on testing their approach and the only fact that could be extrapolated from their conclusions is the fact that time is saved in the RE phase. The rest of the studies are also focused on analysing pattern-based requirements reuse techniques, but none of them is similar to the one presented in PABRE. This fact reinforced the need of conducting a new study in order to validate the PABRE framework.

The study is fully described in the next sections. Section 11.2 describes the research question of the study and the research methodology used. In Section 11.3, the relevant variables that characterize the interviewed professionals of the interviews are stated.

Section 11.4 presents the results obtained, which in some cases are related to the characteristics of the previous section. In Section 11.5 the observations derived from the analysis of the interviews' results are summarized, and, in Section 11.6, the threats to validity of the study are described. Finally, Section 11.7 presents the conclusions of the study.

11.2 Research Approach

The main goal of this study was to investigate in more depth the viability of the use of SRPs and the PABRE framework as a requirements reuse technique, as well as the possible benefits and drawbacks coming from their use. The subsections below provide further details of the methodological approach used.

11.2.1 Research Question

The RQ that drove the study is the following: *Do SRPs, as defined in the PABRE framework, facilitate requirements reuse?* In order to validate the usefulness of SRPs as a requirements reuse technique, it was interesting to ask to the participants their opinion about whether SRPs could enable requirements reuse (i.e. if SRPs could be useful in their organization, how SRPs could be integrated, what pros and cons could emerge from using SRPs, and the critical aspects and barriers for their integration). With the aim of not confusing this RQ with the ones of this thesis, the acronym used in this one is preceded by ES, standing for Empirical Study. Henceforth, this research question will be named ES-RQ. Table 11.1 presents ES-RQ decomposed in different subquestions.

Table 11.1 – Research question of the interview-based empirical study and its subquestions

ES-RQ	Do SRPs, as defined in the PABRE framework, facilitate requirements reuse?
ES-RQ1	Could SRPs be used in organizations?
ES-RQ2	How could SRPs be used in organizations?
ES-RQ3	Are SRPs relevant for all types of requirements?
ES-RQ4	What could be the pros of using SRPs?
ES-RQ5	What could be the cons of using SRPs?
ES-RQ6	What factors could help in the introduction of an SRP catalogue?
ES-RQ7	What factors could represent a barrier for the introduction of an SRP catalogue?

Although the main goal of this study was to validate the PABRE framework from an academic point of view, while carrying out the survey about the state of the practice of requirements reuse and RPs (presented in Section 2.3), it was noticed that there were few empirical studies showing the state of the practice of requirements reuse in industry. Therefore, it was decided to investigate also in these interviews the requirements reuse practices that were being used in organizations. To contextualize the results, it was also studied how requirements were elicited and specified. In this chapter, though, only the results of the main goal of the study are analysed (i.e. the ones related to the SRPs and the PABRE framework), as they are the ones directly related to this thesis.

11.2.2 Research Method

The maturation, acceptance and adoption of new software engineering practices depend on many factors. As Erdogmus states [209], and the results of the survey presented in Section 2.3 point out (see results of the *Critical factors for the successful adoption of an RP catalogue*), the availability of evidence of the benefits in the adoption of a certain practice is one of the critical factors that influence the decision to use this practice in an organization. The aim of Evidence-Based Software Engineering (EBSE) is to encompass these factors [210]. This study was carried out under EBSE principles.

In line with the nature of the research question of this study, a qualitative research approach based on semi-structured interviews to collect data directly from industrial practitioners was chosen [211]. This kind of approach is useful when the purpose is to explore an area of interest, and when the aim is to improve the understanding of a phenomenon [211] [212].

As noticed by Méndez et al. [213], there is a great variability on the way the requirements are defined and handled from project to project. Therefore, the aim was to include practitioners involved in several software development industrial projects from different organizations.

Due to the potential richness and diversity of data that could be collected, semi-structured interviews were considered the most suitable approach for data collection. Semi-structured interviews help to ensure that common information on predetermined areas is collected, but allow the interviewer to go deeper when required. Interviews were chosen over surveys because they allow a better understanding of the questions and a better explanation of the aspect under study (in this case, the PABRE framework). In addition, interviews allow to promote discussions and clarifications when gathering the data, making it possible to elaborate on small aspects the study is investigating and compensate for differences in understanding and terminology, which is very important taking into account that requirements practices and requirements related concepts are very different from project to project. The interview guide used in this study is incorporated in Appendix IV and described in depth in Subsection 11.2.5.

11.2.3 Protocol and Research Team

A research study protocol was designed to register and update the research question, procedures, instruments, decisions and deviations, as suggested by Robson [211]. All the involved researchers participated in the development of this protocol. The design of the protocol, as well as the data collection, was done during the applicant's research stay in Blekinge Technical University (BTH) in Sweden. The researchers' team was formed by the applicant, her advisors and Professor Tony Gorschek (the host research at BTH). Once the research question was formulated, the most suitable methodological approach to answer it was devised. It was decided to survey professionals involved in several software development projects using semi-structured interviews. The data collection and data analysis was carried out by the applicant, and supervised by the rest of researchers.

11.2.4 Sampling

The target population was practitioners in charge of eliciting and/or specifying requirements for software development projects. Participating organizations were chosen from the Swedish industrial network. The selected organizations covered as many different characteristics as possible with respect to size, application domain, and business area. In order to get different views regarding SRPs, it was tried to interview two people from the same organization. A total of 22 interviews were conducted to participants of 11 organizations. Section 11.3 gives further details about the characterization of the participants.

11.2.5 Procedure and Instruments

In order to gather data from the target population, a semi-structured interview guide was carefully designed following the guidelines stated by Oates [214]. In general, the guide focused on a single finished project that the respondents were familiar with. Considering a single project instead of many projects allowed a better interpretation and assessment of contextual information. Otherwise, it would have been very difficult to establish relationships among the considerations of SRPs and the PABRE framework and the characteristics of the project for which the considerations were established. SRPs could have advantages for a certain project of an organization due to some of their characteristics but drawbacks for some other project. The project was chosen by the interviewee without any intervention from the conductors. In addition to this particularization of the inquiries, some follow-up questions were added (such as: *Is this typically how you do this? If not, how do you usually do it?*) in order to identify and understand potential representative practices, as suggested by Lutters and Seaman [215] and Patton [216]. It allowed a richer vision of the requirements processes undertaken by the interviewees and their opinions. The interview guide used in the study may be consulted in Appendix IV.

The interview guide was designed in English. Interviews were performed in English too, which is not the mother tongue neither of the interviewer nor the interviewees. However, being the interviewer mother tongue Spanish and the interviewees' mother tongue Swedish there was no other language that could be used to communicate among them. The only exception was in the case of two interviewees that were Spanish. In those cases, the interviews were carried out in Spanish. Before starting the planned interviews, two pilot interviews were carried out to test the guide and rehearse the interview abilities of the interviewer. In addition, some calibrations of the guide were done after these pilots (mostly related about rephrasing and reordering questions). These pilots were done having as interviewees researchers from BTH that the interviewer did not know, trying to simulate a real interview environment. The resulting guide had 5 sections with the majority of questions being open-ended. Section A and B corresponded to questions about the interviewee's background, the organization and the selected project. Section C and D had questions related to how requirements were elicited and specified in the selected projects, as well as information about requirements reuse (i.e. questions corresponding to a broader objective than the one being analysed in this chapter). Finally, Section E contained questions related to the viability of SRPs, as proposed in the PABRE framework, in the organization at hand (i.e. questions corresponding to ES-RQ).

11.2.6 Data Collection

The interview guide was emailed to each of the interviewees one week before the interview, to allow them to prepare before the interview session. Each of the interviewees was requested to choose a suitable project for answering the interview's questions and to fill in section A of the interview guide (i.e. personal information, organization information and project information). These answers were sent back to the conductors before the interview. Having this information before the interview helped the interviewee to set up their mind on the interviewee's organization's background, and on the selected project. At the beginning of the interview, only if necessary, some clarifications on the organization and/or project at hand were done (corresponding to section B of the guide). Afterwards, data about how requirements were elicited and specified, as well as information about requirements reuse was collected (corresponding to sections C and D of the interview guide). Then, a presentation about SRPs and the PABRE framework was done by the interviewer. Finally, data was gathered about the viability of SRPs in the project and organization at hand (corresponding to section E of the guide).

The interviews were conducted mostly face-to-face, but 5 of them were held using Skype or similar tools. They were carried out by one researcher (the applicant) in English or Spanish (see Subsection 11.2.5 for further information). Interviews lasted around 2 hours each, from which approximately: 45 minutes were dedicated to clarify the organization and/or project (if necessary), to how the requirements were elicited and specified in the project and to whether requirements reuse was applied and how (i.e. the questions of sections B, C and D of the interview guide); 30 minutes were used to do a presentation about the PABRE framework and SRPs, and to solve any possible doubt that the interviewees may have; and the other 45 minutes were used to investigate the viability of SRPs in the organization at hand (i.e. the questions of section E of the guide). The interviews were recorded for subsequent analysis, and notes were taken by the interviewer both during and after the interviews, using a template designed for that purpose, with the main answers provided by the interviewees.

11.2.7 Data Analysis Procedure

The notes taken during the interviews by the interviewer were exhaustively completed with the audio records. These detailed notes, together with the respondents' answers and the audio records, were used during the analysis. The approach followed to analyse the data was coding [217]. To do so, the data of the interviews were analysed using a qualitative data analysis tool called Atlas.ti[38]. Multiple coding techniques were used in different steps:

1. *Attribute* coding was used first to code descriptive information (e.g. interviewer experience, organization domain, organization size, etc.).
2. *Provisional* coding was used to establish a predetermined 'start list' of codes prior to start the coding of the interviews' answers. These codes were defined from the answers of the survey presented in Section 2.3. Therefore, provisional coding was defined for the answers of those questions of the interviews that had an equivalent in the survey, which were: Q4, Q5, Q6, Q7, Q9, Q10, Q12, Q15, Q16, Q21, Q22, Q25, Q26, Q28 and Q29.
3. *Structural* coding and *initial* coding were used to segment the data that relates to a specific subresearch question and question, respectively, of the interview.
4. *Descriptive* coding, *process* coding and *magnitude* coding were used together to code the data from each group identified in the *initia*l coding. *Descriptive* coding was useful to identify the basic topic (what was talked or written about) of a passage of qualitative data. *Process* coding was used to connote specific actions; usually, that was done by identifying codes that corresponded to gerunds ("-ing" words). *Magnitude* coding was used to identify subcodes of the codes coming from the *descriptive* and *process* coding. These subcodes added supplemental alphanumeric or symbolic information to an existing code to indicate its intensity, frequency, direction, presence, etc. Therefore, these subcodes could be qualitative, quantitative and/or nominal indicators. In parallel to these three coding techniques, *simultaneous* coding was used when it was needed to map a statement to two or more different codes or subcodes.
5. *Pattern* and *axial* coding were used to combine similar codes and to establish emerging categories and relationships among them. Specifically, *pattern* coding helped to group under a candidate category the similar activities or factors (i.e. codes) that recurred in the data. *Axial* coding was useful to understand how different categories influence each other, revealing aspects of potential importance.

Steps 1-4 were done by the interviewer and then further discussed with the whole research team in Step 5 to ensure the correct interpretation of each category and the

[38] http://atlasti.com/

evidence that support them. These discussions led to split, modify, discard or add categories to ensure that all the responses and their contexts were well represented. It was tried to be exhaustive with the codes and categories in order to include as much detail provided by the respondents as possible. This assessment was enriched by the information obtained from further questions (such as, *"Is this typically how you do this?"*) that helped to identify and understand practices not used in the particular projects approached, but which might be representative in the organizations. In this way, a broader understanding of the RE practices of each project and organization, as well as of the participants' opinions, was achieved. Frequencies of codes were also generated as an indicator or popular and unpopular practices or opinions. Appendix V presents the association of codes corresponding to the data reported in this chapter.

11.3 Characterization of the Participants, Organizations and Projects

An interview guide was used as the data collection instrument in this study. Its first section contained closed questions aimed to previously gather as much contextual information as possible about the organizations and respondents, in order to understand potential sources of variability, i.e. what works for whom, where, when and why, as stated by Dybå et al. [218]. This information was very helpful to help the interviewer to better prepare the subsequent parts of the interview.

Table 11.2, Table 11.3 and Table 11.4 summarize the main characteristics of the participants, their organizations and the selected projects. In addition, as the aim of this chapter is to analyse the results of ES-RQ (i.e. the one exploring the viability of SRPs, as defined in the PABRE framework, as a requirements reuse technique), the level of requirements reuse that participants carried out in their projects has all been added to Table 11.4, since it was considered that some of the answers of the questions related to ES-RQ may be influenced by this level of reuse. In the following, some aspects of the participants, organizations and projects shown in the tables are discussed.

Participants. As can be seen in Table 11.2, 12 of the participants had an educational background related to computer science, information systems or software engineering; 4 had their background in other types of engineering (such as chemical or civil engineering);

5 had their background related to other areas of science (e.g. telecommunication or robotics); and only 1 did not have an academic background related to these areas but to business. Regarding their highest educational degree, 11 of the respondents had a master's degree, 9 had bachelor's degree and 2 had a Ph.D. degree. The participants had between 3 and 25 years of experience in industry, and between 0 and 15 years of experience in university or research laboratories. The participants had different positions in the organizations, and actively participated in (or were in charge of) RE related processes in at least the project they based their answers on. Some of the participants were new in the position or in the organization (e.g. interviewee K), while others had plenty of experience on both of them (e.g. interviewee F).

Organizations. 11 organizations participated in the study. In 10 of these organizations it was possible to interview more than 1 integrator. Table 11.3 provides an overview of these organizations and relates them to their corresponding respondents. The organizations covered a varied spectrum regarding business areas and size. 5 organizations were software consultancy companies (SCCs) that performed software development tasks for different clients as their primary business; 3 were IT departments (ITDs) in organizations that usually performed or outsource some software development tasks for covering the internal demands of the organization; and 3 were software houses (SHs) that developed and commercialized specific proprietary solutions. In addition, some organizations explicitly stated that their business area was oriented towards a specific domain (stated between brackets in the table). One of the organizations was from the public sector (organization B), and the rest of them were private.

Projects. As explained above, each interviewee was asked to talk about a single finished project. As it can be observed in Table 11.4, the resulting set of projects was very diverse in terms of their domain, their duration and the number of employees dedicated to them. 7 of the projects were embedded systems (projects N, O, P, Q, R, S T), 6 corresponded to websites or mobile applications (projects B, C, D, K, U, V), 6 were focused on systems either to provide a business or to help the customer business (projects G, H, I, J, L, M), and 3 were related to other types of systems (projects A, E, F). Regarding duration and size, the analysed projects took over from 4 months to around 7 years, and they involved from 2 to thousands of people. Only one interviewer (I) did not know the answer to the involved people in the project, and interviewer G made the remark that it depended directly on the project stage. Finally, as for requirements reuse, in 13 projects requirements reuse was inexistent or pretty low (equal or lower than 10%), in 4 it was medium (greater than 10% but equal or lower than 60%), and in 4 it was high (greater than 60%). In addition, in project P the interviewee knew that requirements reuse was applied in the project, but was not sure about the estimated percentage.

Table 11.2 – Characteristics of participants

ID	Highest Educational Background	Years in Industry	Years in University or Research Labs	Job Position	Years in Position	Years in Organization
A	BSc in Computer Science	15	3	Business Analyst	3	3
B	MSc in Computer Science	15	3	Project Manager	≈5	10
C	BSc in Information Systems	20	≈4	System Analyst	6	10
D	BSc in Computer Science	13	3	Requirement Analyst	13	13
E	MSc in Computer Science	25	5	Requirement Analyst	2.5	4
F	BSc in Information Systems	20	0	System Manager	15	20
G	MSc in Computer Science	19	5	System Manager	6	19
H	BSc in Computer Science	15	0	Senior Project Manager	15	15
I	BSc in Energy Systems	20	0	Senior Business Consultant	6	6
J	MSc in Computer Science	16	0	Senior Developer	9	9
K	MSc in Software Engineering	17	5	Consultant Manager	0	0
L	MSc in Business	12	≈5	Solution Designer	≈8	≈10
M	BSc in Computer Science	23	0	Business Analyst	14	14
N	PhD in Food Engineering	10	15	System Engineer	2	5
O	MSc in Chemical Engineering	10	0	System Engineer	0.25	7
P	BSc in Telecommunication	25	0	Product Manager	5	19
Q	MSc in Industrial Engineering	8	0	System Engineer	8	8
R	MSc in Computer Vision and Robotics	9	5	Project Leader	2	2
S	MSc in Electrochemistry and Electronic Sensors	3	3	Lead Engineer	0.5	2
T	PhD in Civil Engineering	23	10	Software, Manufacturing & Electrical Engineer	1.5	16
U	MSc in Computer Science	21	0	Senior Consultant	5	12
V	BSc in Interaction Design	9	3	Senior Consultant	3	9

Table 11.3 – Characteristics of organizations

ID Organization	ID Respondent	Number Employees	Main Business Area
A	**A, B**	≈2,000 WW	ITD of a Telecommunication Operator
B	**C, D**	≈900	SCC of the Public Sector
C	**E**	≈350	SH (UI Platforms for Symbian-Based Smartphones)
D	**F, G**	≈800	SH (Telecommunication Products)
E	**H, I**	≈68,000 WW	SCC
F	**J, K**	50	SCC
G	**L, M**	800	SCC (Telecommunication Products)
H	**N, O**	≈23,000 WW	ITD of a Tetrabriks Manufacturer
I	**P, Q, R**	≈150,000 WW	SH (Power and Automatization Systems)
J	**S, T**	≈20,000	ITD of a Car Manufacturer
K	**U, V**	1,200	SCC

Table 11.4 – Characteristics of projects

ID Project	ID Respondent	Project Main Functionality	Project Domain	Project Duration (in years)	Project Number Employees	Project Reqs. Reuse (%)
A	**A**	Getting customer feedback	Messaging System	1	≈10	0%
B	**B**	Webshop for acquiring phones and contracts with a carrier	Website	1	≈10	0%
C	**C**	Translating to English a website	Website	1	10-12	10%
D	**D**	Management of the social security rights of children	Website	1.5	≈35	70-75%
E	**E**	OS for a specific smartphone taking into account the carrier's restrictions	Mobile OS	0.5	≈100	≈40%
F	**F**	Carrier system to track the users' consumption	Machine to Machine System	0.25	7	0%
G	**G**	Providing services to customers (charging, changing plan, consumption, etc.)	Carrier Business Support System	2.5	15-500 (depending on the stage)	0%
H	**H**	Managing consumption energy levels measured by energy organizations	Energy Measurement System	1.5	≈2	10%
I	**I**	System for an energy company involving the contract and offering module	Business Support System	2	Not sure	0%
J	**J**	System for a carrier involving big data, call data management, contracts management, etc.	Carrier Internal System	1	100	0%
K	**K**	Webshop for acquiring public transport system tickets	Website	0.33	5	10%
L	**L**	Offering roaming services to customers	Carrier Business Support System	≈1.5	≈20	0%
M	**M**	Managing customer calls into the customer service centre	Carrier Internal System	1.5	25	0%
N	**N**	Modifying an existing machine (and its internal system) to make it more productive	Embedded System	4	35	15-60%
O	**O**	New machine (and internal system) for a new package	Embedded System	0.75	10	80%
P	**P**	Managing control and safety processes on fabrics	Embedded System	≈1.5	≈200	Reuse applied, Not sure %
Q	**Q**	Controlling the machines of a sugar fabric	Embedded System	1.5	6	5-10%
R	**R**	Managing the different functionalities of a car	Embedded System	≈3	≈60	30-40%
S	**S**	Controlling the charge of battery in electric cars	Embedded System	2	20	60-70%
T	**T**	Controlling the machines for producing a car	Embedded System	6-7	x000	80-85%
U	**U**	Checking films, book tickets, etc. for a cinema company	Mobile App	1	18	≈20%
V	**V**	Integrating payment services	Mobile App, Website	0.25	12	0%

11.4 Results

The preliminary results of ES-RQ, which is related to SRPs and the PABRE framework, are presented in this section. In order to answer it, participants were asked about seven different aspects, each one of them related to a subresearch question: whether SRPs could be useful in the participants' organization, how SRPs could be used there, for what types of requirements SRPs would be useful, what pros and cons could emerge from using SRPs, and the critical aspects and barriers for their integration.

ES-RQ1 Could SRPs be used in organizations? Participants were asked their opinion about whether SRPs would enable reuse in their organization. As can be seen in Table 11.5, only one participant stated that SRPs could not be used in the organization (interviewee A). When asked the reason behind that opinion, the answer was related to the way of working in the projects the respondent usually dealt with: "*Because of the way of working in the projects. There is no established level of abstraction of the requirements in the specifications (sometimes it is what to do, sometimes how to do it), so it will be difficult to establish a set of SRPs to reuse. Usually I work most of the time as a consultant, and it is not clear who is in charge of the requirements sometimes: the organization or the customer.*" (A). However, the respondent added that in other types of environments, SRPs could be useful: *"I see the use of it [the SRP approach] in other projects that are more structured. I see the potential of SRPs, but I think our projects are not mature enough (in terms of requirements) as to be using them."* (A). In addition, it is worth remarking that this interviewee did not carry any requirements reuse in the selected project, which supports the fact of not being able to use SRPs.

The rest of the participants agreed that SRPs could be used in their organizations. The main reasons behind that opinion are shown in Table 11.6. The most recurrent ones are:

1. Specifics characteristics of the SRPs, e.g. different set of requirements (i.e. clusters) are provided in an SRP, the incorporation of a glossary of terms, and the existence of parameters and their associated domains (9 participants).
2. The use of SRPs would enhance the quality in the resultant SRSs (9 participants)
3. The fact that SRPs incorporate reusable requirements (8 participants). Related to this reason, participants specially highlighted the fact that SRPs would define a base of requirements ready to reuse: *"They [SRPs] provide a basis of requirements ready*

to reuse. This would not solve all the problems related to requirements, but for the requirements specification and the requirements writing, it would help a lot." (G).

There are other factors that, although being less frequent, are worth to highlight. Regarding the organizational factors, respondents stated that SRPs would enable reuse since they usually have similar requirements among projects, but usually it is necessary to do small changes on the requirements (what would correspond to the parameters in the SRPs). Other participants also stated that SRPs would bring benefits in the whole project lifecycle. Specifically, they highlighted that SRPs would make the RE stage less people dependent, and that SRPs would help to trace requirements in the project lifecycle. Finally, a respondent answered that he could not see any reason for SRPs not being useful: "*Yes, I see them useful. I do not see why they couldn't be used, I do not think it is anything strange.*" (D).

Table 11.5 – Categories of the responses to whether SRPs would enable reuse

Code	Description	Respondents	Frequency	Percentage
Use-Yes	**Yes**: SRPs would enable reuse in the respondents' organizations.	B, C, D, E, F, G, H, I, J, K, L, M ,N, O, P, Q, R, S, T, U, V	21	95%
Use-No	**No**: SRPs would not enable reuse in the respondents' organizations.	A	1	5%

Table 11.6 – Categories of the reasons that explain that SRPs would enable reuse

Code	Description	Respondents	Frequency	Percentage
Use-A	**SRP specific characteristics**: SRPs have certain characteristics that allow them to be used in the organizations.	B, D, J, P, Q, S, T, U, V	9	41%
Use-B	**Better SRSs quality**: the use of SRPs would drive to improve the quality of SRSs, which is a desired aspect by the respondents.	C, E, F, I, K, L, M, R, T	9	41%
Use-C	**SRPs incorporate reusable requirements**: SRPs define requirements that will be reused in other projects.	E, G, H, I, L, M , P, S	8	36%
Use-D	**SRPs are easy to use**: SRPs are considered easy to use for requirements reuse.	A, F, H, O, P	5	23%
Use-E	**SRP organization**: SRPs are organized by means of different classification schemas, allowing to choose the one that is more adequate for an organization.	K, N, P, V	4	18%
Use-F	**Organizational factors**: there are specific factors from the organizations that allow to use SRPs.	A, P, S	3	14%
Use-H	**SRPs improve project lifecycle**: SRPs have consequences not only in the RE stage, but also in others stages of a project lifecycle.	B, I, K	3	14%
Use-I	**SRPs are flexible during their use**: while using SRPs, modifications on them can be done, so the reuse process is not rigid.	G, U	2	9%
Use-J	**Why not?**: the respondents stated that they did not see any reason for not using SRPs.	D	1	5%

ES-RQ2 how could SRPs be used in organizations? All the participants that answered positively to the previous question (i.e. if SRPs could be used in their organizations) were asked about how they imagined that SRPs could be used in their organizations. Table 11.7 shows the different responses provided by the participants. Some of the most frequent categories are:

1. Using SRPs by searching and selecting them (14 participants). 8 from the 14 participants supporting the search and selection of SRPs considered that it would be necessary to have a database of SRP to support the process.
2. Using SRPs during the elicitation and specification of requirements (9 participants).

Table 11.7 – Categories related to how SRPs could be used in the organization

Code	Description	Respondents	Frequency	Percentage
Inte-A	**Searching and Selecting SRPs**: specific SRPs could be searched and afterwards selected to use them in projects.	B, C, E, G, H, K, L, P, Q, R, S, T, U, V	14	64%
Inte-B	**Only for specific requirements or projects**: the respondents highlighted the fact that SRPs could only be integrated for certain types of requirements or projects in their organizations.	B, C, D, F, G, J, K, L, N, P, R, U	12	55%
Inte-C	**During elicitation and specification**: SRPs could be used during the elicitation and specification of the requirements for a project.	B, G, J, K, L, O, P, Q, U	9	41%
Inte-D	**Having an SRP database**: respondents considered necessary to have a database of SRPs to be able to integrate SRPs in their organizations.	B, P, Q, R, S, T, U, V	8	36%
Inte-E	**As a checklist**: SRPs could be used to check that no requirement has been forgotten.	B, C, I, K, M, R, S	7	32%
Inte-F	**Define templates based on SRPs**: SRPs could be used to define a predefined set of requirements that need to be used, which could be incorporated in templates of SRSs.	F, N, O	3	14%
Inte-G	**Other RE activities**: SRPs could be used in other RE activities different from elicitation and specification (such as for the decomposition of high-level requirements).	B, C, F	3	14%
Inte-H	**Bulk specification**: SRPs could be used in a less driven manner than the ones proposed in the PABRE framework.	V	1	5%
Inte-J	**Cross-referencing SRPs**: the use of an SRP could lead to another one directly.	H	1	5%
Inte-K	**Defining high-level constants**: SRPs could be used defining first the value of high-level project's constants (e.g. the currency that want to be used in a project), defining in this way at once the value of all the parameters in all the SRPs that are using this value.	E	1	5%

The respondents also considered necessary to highlight here that SRPs could only be used for certain types of recurrent requirements or for projects that share very similar requirements (12 participants). This was specially remarked in the case of interviewees coming from SCCs (6 out of the 10 respondents that belonged to an SCC remarked it), probably because their types of projects were usually quite different: *"They [SRPs] would be useful for small project customizations that are similar."* (J).

A less common response is the use of SRPs with a checklist character (7 participants), with the aim of not forgetting any requirement: *"As a checklist, to make sure they [the requirements analysts] have cover all the aspects of the system."* (C). Other interesting ways of using SRPs only highlighted by one participant are: using SRPs in a more bulk manner: *"Searching for a previous project and checking the SRPs you want to use from that project, and maybe the value of the parameters that have been used in that project too."* (V); and defining values for high-level parameters: *"You can also use them to define high level values of the parameters that could be used in a project, so you change that value in one place, and all the requirements that use this parameter change also their value."* (E).

ES-RQ3 Are SRPs relevant for all types of requirements? The participants were asked whether they thought that SRPs would be relevant for all types of requirements and, even if the participants answered positively to the previous question, for what specific types of requirements SRPs would be relevant. Table 11.8 shows that only 3 participants thought that SRPs could be relevant for all requirement types. The main reason behind that opinion was that the interviewees could not think about any requirement type for which SRPs could not be used: *"Yes, I can't see why patterns could not work for all types."*. However, the reason for respondents stating that SRPs could not be used for all types of requirements (19 participants) was more in the line that it does not have sense to convert all requirements into SRPs. *"No, because there are things that are too specific for a project (for instance, the user interface requirements), and these specific things cannot be reused."* (E).

As for the types for which SRPs could be relevant, Table 11.9 contains the results. The most important facts that could be highlighted here are:

1. Most of the participants agreed that SRPs could be used for NFRs (18 respondents) and FRs (14 respondents).
2. In the case of FRs, most of the participants thought SRPs could only be used for those requirements that are common in a functional domain (e.g. CMS) or if the

customer is the same: "… *[SRPs would be useful for] some functional requirements that are quire repetitive*" (D), "… *for the functional requirements, they [SRPs] would be especially useful if the customer is the same, because if the projects or customers are too different probably the functional requirements are also too different.*" (Q).

3. Only 4 of the participants that stated that SRPs were not relevant for all types of requirements mentioned that SRPs could be used for NTRs (interviewees E, H, M, T).

Table 11.8 – Categories of the responses to whether SRPs would be relevant for all types of requirements

Code	Description	Respondents	Frequency	Percentage
ReqTy-Yes	**Yes**: SRPs would be useful for all types of requirements.	A, C, S	3	14%
ReqTy-No	**No**: SRPs would not be useful for all types of requirements.	B, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, T, U, V	19	86%

Table 11.9 – Categories of types of requirements for which SRPs would be or would not be relevant

Code	Description	Respondents	Frequency	Percentage
ReqTy-A	**NFRs**: SRPs would be relevant for non-functional requirements.	A, B, C, D, E, F, G, H, J, K, L, M, P, Q, S, T, U, V	18	82%
ReqTy-B	**FRs**: SRPs would be relevant for functional requirements, especially for those requirements that are common in a functional domain or if the customer is the same.	A, B, C, D, E, F, H, J, K, M, Q, S, T, V	14	64%
ReqTy-C	**NTRs**: SRPs would be relevant for non-technical requirements.	A, C, E, H, M, S, T	7	32%
ReqTy-D	**Specific level of requirements**: SRPs would be relevant only for specific levels of abstraction of requirements.	I, L, N, O	4	18%
ReqTy-E	**Other types of requirements**: SRPs would be relevant for other types of requirements that are very specific of the organization.	O, R	2	9%
ReqTy-J	**There are types of requirements for which SRPs would not be relevant**: the respondents stated that there are some requirements for which SRPs would not be relevant (e.g. creative requirements).	D, E, F, H, I, J, K, L, M, N, O, P, Q, T, U, V	16	73%

Some participants also stated that SRPs could be used only in some specific levels of requirements. This was especially relevant in the case of interviewees N and O, since their requirements were always divided in three different levels (stakeholder, system and subsystem requirements): "*SRPs would only be useful for the common ones, the ones at the subsystem level.*" (N). In addition, 2 participants stated that SRPs could also be used in other types of requirements that were difficult to classify as FRs, NFRs or NTRs: they were talking about the requirements coming from standards that needed to be fulfilled (interviewee O) and the requirements in the *diagnostic area* section of SRSs (interviewee R). Finally, most of the participants (16 respondents) stated that there were specific types of requirements for which SRPs would not be relevant. Some of the types stated there are: creative requirements (i.e. the ones that specify the creative or different part of a system), user interface requirements and computational requirements.

ES-RQ4 What could be the pros of using SRPs? The participants were asked about their opinion on what would be the benefits of using SRPs (Table 11.10). Most of these benefits are related to the quality of the resultant SRS or product (9 out of 17 benefits: Pro-B, Pro-C, Pro-D, Pro-E, Pro-F, Pro-I, Pro-J, Pro-L, and Pro-M) or to the reduction of the project's resources spent (Pro-A, and Pro-P). The other pros are related either to improving other aspects of requirements (understanding in the case of Pro-G, and traceability in the case of Pro-N), to improving certain activities (the whole lifecycle in the case of Pro-H, and the requirements management in the case of Pro-K), to improving the definition of high-level constants in the projects (Pro-O), and to improving the definition of the stakeholders' needs (Pro-Q). Participants specially agreed on the following five pros coming from the use of SRPs:

1. Less time would be spent during the elicitation and specification of requirements (17 participants), as SRPs provide ready-to-use requirements: "*Less time would be needed for eliciting the requirements, because you have requirements in which you can search, you do not start from scratch.*" (Q).
2. Better quality in general in the resultant SRSs (15 participants), as SRPs contain requirements that are of good quality: "*The first one [benefit] is of course the better quality of the specifications, because the requirements at SRPs would be well written and so on.*" (V).
3. There would be less missing requirements (10 participants), as SRPs could be used as a checklist to not forget requirements: "*Requirements patterns could help with the incompleteness of the specifications, because there are requirements that are implicit and that are usually forgotten, and having the patterns would help to make them explicit and have them into account from the start.*" (L).
4. Requirements would be less ambiguous (8 participants), because the templates in the SRPs have been checked against ambiguity: "SRPs would help with the

ambiguity of requirements, because if SRPs are already stated clearly, this helps to put stakeholders' views on the same page and to establish a base ground of common knowledge." (O).

Table 11.10 – Categories of the pros that would come from using SRPs

Code	Description	Respondents	Frequency	Percentage
Pro-A	**Less time spent in elicitation and specification**: the use of SRPs could drive to spending less time in both the elicitation and specification of requirements.	A, B, C, D, F, H, I, K, L, M, N, O, Q, R, S, U, V	17	77%
Pro-B	**Better SRS quality**: the use of SRPs could drive to having more quality in the resultant SRSs.	C, E, F, H, I, J, K, L, M, P, R, S, T, U, V	15	68%
Pro-C	**Less missing requirements**: the use of SRPs could drive to missing less requirements in the resultant SRSs.	B, E, H, I, K, L, M, Q, U, V	10	45%
Pro-D	**Less ambiguous requirements**: the use of SRPs could drive to having natural language requirements that are less ambiguous in the resultant SRSs.	A, E, I, M, N, O, P, T	8	36%
Pro-E	**More uniform requirements**: the use of SRPs could drive to having requirements that are more uniform in the resultant SRSs.	A, D, F, G, H, I, T	7	32%
Pro-F	**More product quality**: the use of SRPs could drive to having final products or systems of better quality.	Q, R, S, U, V	5	23%
Pro-G	**Better requirements understanding**: the use of SRPs could drive to requirements that are easier to understand.	F, J, M, T	4	18%
Pro-H	**Better project lifecycle**: the use of SRPs could drive to improving not only the RE stage, but also others stages of a project lifecycle.	B, I, K	3	14%
Pro-I	**Less implicit information in the requirements**: the use of SRPs could drive to requirements that have less implicit information in the resultant SRSs.	A, C, F	3	14%
Pro-J	**Less inconsistent requirements**: the use of SRPs could drive to requirements that are less inconsistent in the resultant SRSs.	C, E, H	3	14%
Pro-K	**Better requirements management**: the use of SRPs could drive to managing requirements in an easier way.	A, C	2	9%
Pro-L	**Better requirements writing**: the use of SRPs could drive to specifying requirements that are written better.	P, T	2	9%
Pro-M	**Better requirements organization**: the use of SRPs could drive to requirements that are better organized in the resultant SRSs.	E	1	5%
Pro-N	**Better requirements traceability**: the use of SRPs could drive to improving the traceability of the requirements in the project lifecycle.	G	1	5%
Pro-O	**Easy to define high-level constants in the projects**: the use of SRPs could drive to defining constants in the projects (such as the currency) in an easy way.	A	1	5%
Pro-P	**Less money spent**: the use of SRPs could drive to reduce the costs of projects.	H	1	5%
Pro-Q	**Better stakeholders' needs**: the use of SRPs could drive to having the needs of the stakeholders better established.	O	1	5%

5. Requirements would be more uniform (7 participants), as the same structure has been used to write all the SRPs: "*SRPs would help with the uniformity of the requirements, as all SRPs would be stated in the same way.*" (D).

ES-RQ5 What could be the cons of using SRPs? The participants' opinion were asked about the possible drawbacks that would come from using SRPs. Table 11.11 contains the categories found in the answers of that question. Some participants could not think of any drawback that may come from using SRPs (6 participants), and one of them specifically stated that it was difficult to know what cons SRPs would have if they were not using SRPs: "*I'm not sure… as I'm not using them [the SRPs] it is difficult to know any drawback.*" (E). Another interviewee stated that there would be much more benefits than drawbacks while using SRPs: *"I'm sure that the benefits are larger than the possible drawbacks."* (B).

The possible cons of SRPs stated in Table 11.11 (from Con-D to Con-P) are of different nature. The ones stated from more than one respondent are:

1. People getting too used to SRPs (7 participants), in such a way that SRPs would end up not being used properly: *"Also getting lazy on thinking on the requirements, because you could get too used to use them in a bad way, only use the patterns because they are in the catalogue, but not checking them out."* (L).
2. The difficulty to create the right SRP catalogue (7 participants), as it is difficult to know what requirements shall be converted into SRPs and which ones shall not: *"The catalogue has to have the right content, otherwise it is not useful, and that is not easy to do."* (C).
3. Having less creative products (6 participants), as users of SRPs could think that all the solutions would be in the SRP catalogue: *"One of the drawbacks is that people might get less creative: SRPs should not be considered like they are a close set of requirements, and that all the requirements would be there. Otherwise, people would lose their imagination and the products would be less and less innovative."* (N).
4. The maintainability of the SRP catalogue (5 participants), as there would be a considerable quantity of knowledge that would need to be maintained: *"Another drawback is the amount of knowledge needed to maintain, the difficulty and the time needed to create the SRP catalogue."* (J).

Table 11.11 – Categories of the cons that would come from using SRPs

Code	Description	Respondents	Frequency	Percentage
Con-A	**None**: the respondents could not think about any drawback that may come from the use of SRPs.	D, F, I, P, T	5	23%
Con-B	**Difficult to know**: the respondents stated that it was difficult to know any drawback that could come from the use of SRPs if they are not actually using SRPs.	E	1	5%
Con-C	**More benefits than drawbacks**: the respondents stated that although could exist drawbacks from using SRPs, the benefits SRPs may provide are larger.	B	1	5%
Con-D	**Get too used**: people using SRPs could end getting too used to SRPs, in such a way that they will not use SRPs properly.	B, K, L, M, N, O, Q	7	32%
Con-E	**Creating the right SRP catalogue**: it could be difficult to know what requirements shall be converted into SRPs and which ones shall not.	B, C, J, K, R, S, V	7	32%
Con-F	**Less creative products**: the use of SRPs could drive to having final products or systems that are less creative.	B, H, K, N, O, Q	6	27%
Con-G	**Maintainability of SRPs**: there are aspects associated to the maintainability of SRPs that could be difficult to take care of.	B, C, G, J, U	5	23%
Con-H	**Completeness of the SRP catalogue**: the catalogue of SRPs would never be complete.	K, L, M, Q	4	18%
Con-I	**People-related factors**: difficulties to introduce SRPs in an organization because, at the end, humans would be the ones using them.	A, H	2	9%
Con-J	**Bad quality of SRPs**: although SRPs are supposed to have good quality, ultimately that depends on the people creating/maintaining them, and if SRPs end having bad quality, this bad quality or errors would be replicated.	M	1	5%
Con-K	**More time needed for the RE stage**: the elicitation and specification of requirements could become heavier because of the use of SRPs, and more time would be needed.	M	1	5%
Con-L	**Learning curve**: it is necessary to use SRPs for some time before someone would know how to use SRPs properly and efficiently.	J	1	5%
Con-M	**New tool**: the use of SRPs would imply incorporating a new tool into the organization.	A	1	5%
Con-N	**Being constant**: to get benefits from using SRPs, it would be necessary to being constant and continue working with SRPs, even if at the beginning no benefits are achieved.	G	1	5%
Con-O	**Vision of SRP as 'boring'**: people using SRPs could think that SRPs are boring, and therefore they would not want to work with them.	G	1	5%
Con-P	**Integration with agile methodologies**: it could be difficult to integrate SRPs with more agile methodologies.	V	1	5%

5. The completeness of the SRP catalogue (4 participants), again because SRP users would think that all the requirements would be in the catalogue: *"People may think it [the SRP catalogue] is complete, then mistakes will be done and things will be missing."* (K).

6. Factors related to the people using SRPs (2 participants), as it is difficult to change the way people works: *"Putting all the people to work on the same way, so you use them properly and get benefits."* (A) and *"The technical parts [for using SRPs] are easy to establish, but the human part is always difficult."* (H).

From the drawbacks stated by just one respondent, it is worth to highlight the one that talks about more time being needed for the RE stage because of the use of SRPs. The interviewee pointed out that although more time might be needed for the RE stage, SRPs would help to save time when looking at all the time spent in a project: "*…although the time might be bigger by using SRPs because you add an extra step to check for completeness, at the end you save time by not missing requirements.*" (M).

ES-RQ6 What factors could help in the introduction of an SRP catalogue? Participants were asked about what factors they thought that would help for introducing SRPs in their organization. Table 11.12 shows the results of this question. The most frequent introduction factors are:

1. Training courses on SRPs in the organizations (16 participants), as they would help to teach how to use SRPs. Although most of the participants agreed that these courses shall be taught to all the organization's members that are going to work with SRPs, interviewees C and O thought that maybe it would be better to teach them only to specific members or team heads: *"Having training courses, but only for the right people, to the people that will use them or to the team head, so not everyone lose their time."* (O).
2. Showing the costs and benefits coming from the integration and use of SRPs (14 participants), as this would help to engage people to use SRPs. Most of them (13 participants, all except interviewee P) thought that a good idea for showing the costs and benefits would be trough success cases of SRPs: "*Having success cases of other people using SRPs, because that would make people more interested in starting using them.*" (H). In addition, 3 of the participants (interviewees N, Q, and R) highlighted here that it is important to quantify these benefits not only in terms of money, but in terms of production and quality: "*Seeing if SRPs are economically worth. […] I meant, it would be good to have the numbers on how good SRPs are, not only regarding money but in terms of quality and production too.*" (N).

3. A tool that supports the SRP use and maintenance (13 participants), as this would ease the use and maintenance of SRPs. Interviewees L, K and V remarked that it is important that the tool is a good tool and easy to use: *"The existence of a good tool support. According to my experience, what I have seen, the simpler the tool for using SRPs is, the better."* (L).
4. An SRP catalogue that is ready to be used by organizations (11 participants), to facilitate the first use of SRPs in the organizations: "*Also having a ready-to-use SRP catalogue (especially for a first typical domain in the organization), to be able to train and show it to people, and to have a basis to start with.*" (J).

Table 11.12 – Categories of the introduction factors that could help during the introduction of SRPs

Code	Description	Respondents	Frequency	Percentage
Intr-A	**Training courses**: having training courses to teach users of SRPs how to use them.	A, B, C, D, F, G, I, J, L, N, O, P, Q, R, T, U	16	73%
Intr-B	**Showing costs and benefits**: being able to show what are costs of integrating and maintaining SRPs, as well the benefits that using SRPs would provide.	B, C, D, H, I, J, N, O, P, Q, S, T, U, V	14	64%
Intr-C	**Tool support**: having a tool that eases the use of SRPs and its maintenance.	E, F, G, I, J, K, L, N, P, Q, S, U, V	13	59%
Intr-D	**Ready-to-use SRPs**: having SRPs that are ready to be used by organizations.	B, D, E, H, I, J, K, M, R, S, T	11	50%
Intr-E	**Exemplification**: having examples or demos that are relevant for the organization where SRPs will be introduced.	B, C, D, I, J, K, P, Q	8	36%
Intr-F	**Organization's expert in SRPs**: having someone in the organization that is an expert on SRPs.	B, N, O, R, S, U	6	27%
Intr-G	**Usage method**: having a well-defined method to use SRPs.	A, K, L	3	14%
Intr-H	**People involvement**: encouraging people to commit to use SRPs.	T	1	5%
Intr-I	**Integration of SRPs in organization's tools**: having tool support for SRPs, but integrating it with a plugging in the tools already used by the organization.	S	1	5%
Intr-J	**Starting small**: introducing SRPs at first only in some specific small project, and growing from there until SRPs are totally integrated in the organization.	U	1	5%

Other factors less common among the interviews' answers, but still agreed by some of the participants, are: 1) The exemplification of SRPs or having demos about them (8 participants), specially for a typical or relevant domain in the organization (5 participants; interviewees C, J, K, P, and Q): *"Attractive examples of SRPs relevant for the organization and the type of people you are talking to (marketing, managers, coders, analysts, etc.), because that would make people see that SRPs are attractive for their organization."* (C). 2) Having an expert on SRPs in the organization (either a single person or a department) (6 participants), so not every SRP user would need to know the aspects related to the management of SRPs: "*[…] and the existence of an organization's person or department expert on SRPs, so not everyone has to know how to manage all the knowledge to be reused.*" (O).

Finally, interviewee U made a good point about how SRPs shall be integrated in their organization: "*Keeping it [the integration] as simple as possible. It is better to start small, and then enlarging the use of SRPs, so their introduction is easier.*" (U).

ES-RQ7 What factors could represent a barrier for the introduction of an SRP catalogue? The respondents gave their opinion about the factors that could hinder the introduction of SRPs in their organization. The categories extracted from the interviews' answers related to this matter are presented in Table 11.13. One of the interviewees stated that there would be no barriers provided people see the benefits of SRPs: *"I don't think that changing the way of working would be a problem, provided people understand patterns easily and they see the benefit of using them."* (B). Interviewee H also pointed out that there would be always barriers when introducing a new technique: "*[…] there are always barriers when introducing something new.*" (H). The three most common barriers stated by the interviewees are:

1. The integration with the existent RE processes of organizations (14 participants), as changing the way organizations works is usually difficult: "*Well… The first drawback would be the integration with the existing processes, because changes are always difficult.*" (J). Related to this barrier, 4 of the interviewees (F, N, O, and V) were especially worried about how to integrate SRPs properly taking into account how the organization worked: "*Yes, the integration with the existing RE processes. Because big organizations usually have large requirement processes, and then different people is in charge of requirements, and at different levels, and there is no direct communication among them. These people might even be spread away, even in different countries. So changing that way of working in such situation is not easy.*" (F). The reason behind that could be that the organizations of these interviewees had very rigid and well-defined RE process, which could make especially difficult the integration of a new technique.
2. The resistance of people to change their way of working (13 participants), as changing the way humans do things is always difficult: "*[…] and the resistance of*

requirement engineers to change, because they are used to work in one way, so even if they see the benefits, changing is always tedious. People may not want to use a new thing." (Q).

3. The maintenance of the SRP catalogue (7 participants), as respondents considered especially difficult to manage a big amount of reusable knowledge and they were not sure how the maintenance process would be carried out in their organizations: "*Another barrier is the amount of reusable knowledge necessary to maintain. How the knowledge will be created and maintained? It is massive the work you need to establish SRPs as a reuse technique in the organization, because we manage a huge amount of requirements and products, so it is difficult to create that much SRPs and maintaining them.*" (N). Tough, interviewee J highlighted that this maintenance is unavoidable in any reuse technique: "*It is true that the amount of reusable knowledge necessary to create and maintain is huge, but that is unavoidable with any reuse technique.*" (J).

Table 11.13 – Categories of the barriers that could hinder the introduction of SRPs

Code	Description	Respondents	Frequency	Percentage
Bar-A	**None**: the respondents could not think about any barrier that could hinder the introduction of SRPs.	B	1	5%
Bar-B	**Always barriers**: the respondents stated that there are always barriers when introducing a new technique in organizations.	H	1	5%
Bar-C	**Integration with organization's processes**: difficulties that could arise when integrating SRPs with the existent RE processes of the organizations.	A, D, E, F, G, H, J, K, L, N, O, Q, T, V	14	64%
Bar-D	**Resistance to change**: difficulties related to the fact that people would have to change their way of working because of the introduction of SRPs.	A, C, D, E, H, J, M, O, Q, R, S, T, U	13	59%
Bar-E	**Maintenance**: difficulties related to the maintenance of SRPs and the knowledge needed.	C, G, J, N, P, U, V	7	32%
Bar-F	**Cost and effort**: difficulties related to the cost and effort of introducing SRPs in organizations.	G, P, V	3	14%
Bar-G	**Management support**: difficulties related to the lack of management support for introducing SRPs.	A, I	2	9%
Bar-H	**Heavier RE stage**: difficulties related to the fact that the elicitation and specification of requirements could become heavier because of the use of SRPs.	G, I	2	9%
Bar-I	**Vision of SRPs as 'one more thing'**: difficulties related to the fact that people using SRPs may think that SRPs are one more thing to use, and therefore they would not want to work with them.	K	1	5%
Bar-J	**Full integration to get benefits**: difficulties related to the fact that SRPs need to be fully integrated in an organization in order to get benefits.	A	1	5%
Bar-K	**Academic wording**: difficulties related to the fact that the wording used in the tool/plugging could end being too academic.	K	1	5%

11.5 Discussion of the Results

This section includes the observations derived from the analysis of the interviews' results. A comparison with the results of the survey-based empirical study presented in Section 2.3 is also included in each observation.

SRPs would enable requirements reuse in different types of organizations. Most of the participants agreed that SRPs would enable reuse in their organizations (Table 11.5). The only respondent that did not agree stated that the cause was related to the difficulty of establishing a set of requirements to reuse in their projects (because the way of working with requirements was not mature enough: there was not a clear level of abstraction of the requirements neither it was clear who was the main responsible of the requirements) rather than to the SRP approach per se. The participants that stated that SRPs would enable requirements reuse represented a rich variety in terms of organization size, organization main business area, project size and project domain (see Section 11.3 for further details). In addition, it is worth remarking that most of the respondents stated to not have carried out requirements reuse in their projects or that it was of a very low level (55%, see Table 11.4), which makes even more relevant the fact that they believed that SRPs would enable requirements reuse in their organizations.

This variety allows to corroborate the fact that SRPs could be used in different types of organizations and projects, provided the RE processes are well-established on them. In addition, the conclusion is corroborated by the answers to the survey's questions related to RQ 1.2.2 (Subsection 2.3.5), where some participants who declared an inexistent or very low level of reuse stated that it was related to the lack of process maturity in organizations, as stated by the interviewee that answered negatively to the question about SRPs enabling requirements reuse.

SRPs would be used mainly by searching and selecting SRPs or as a checklist, during the elicitation and specification of requirements. As seen in Table 11.7, most of the participants agreed that SRPs would be used in their organizations during the elicitation and specification of requirements by searching SRPs and afterwards selecting the ones that want to be used in a project. 36% of respondents thought that this search process should be supported by a database with SRPs. The use of SRPs with a checklist character is the second most agreed approach by the participants (32%).

SRPs would be relevant for NFRs and recurrent FRs among projects. As it was expected, participants considered that SRPs would not be useful for all types of requirements (Table 11.8, and Inte-B in Table 11.7). With respect to the type of requirements for which SRPs would be relevant, NFRs and recurrent FRs are the ones that participants believed to be more relevant (Table 11.9). The interpretation is that NTRs are not necessary in all IT projects (usually they are only included in call-for-tender projects), or when they are included, they do so as part of NFRs, and therefore only those respondents that usually used NTRs considered that SRPs would be relevant for this requirement type. These results align with the results of the survey-based empirical study, were NFRs was the type considered to be most recurrent, closely followed by FRs (Subsection 2.3.4).

Benefits achieved by the use of SRPs would be mainly related to the time spent for the RE process and the quality of the resultant SRSs. The two benefits that could be achieved by the use of SRPs would be related to spending less time for the elicitation and specification of requirements and to the quality of requirement specifications (Table 11.10). Regarding time saving, the results are a little bit different from the ones achieved in the survey-based empirical study, where only the participants with experience in RPs believed that RPs could help to spend less time in the elicitation of requirements (see Subsection 2.3.6). The reason behind that is probably that in the current study, participants got a complete presentation about SRPs and the PABRE framework, in contrast to the survey participants, who only got a small explanation about what RPs are. Therefore, the respondents of the interviews, as well as the respondents of the survey that had previously used RPs, were better informed to have a more clear idea of what the benefits of requirement patterns could be.

As for the improvement of the quality of the resultant SRSs, this is a logical consequence of working with a knowledge base of reusable artefacts that is supposed to contain artefacts with certain quality. In addition, some of the interviewees stated specific quality aspects that could be improved, being the most frequent ones: the completeness of SRSs, the requirements uniformity, and the requirements ambiguity. These benefits are completely corroborated by the survey results, where these three aspects were ranked among the most common problems to be mitigated by RPs (Subsection 2.3.6). The respondents of the survey also ranked as one of the most common problems to be mitigated by RPs the inconsistency of requirements. This answer also appears in the current study, but only three interviewees stated it.

Critical factors for the successful adoption of SRPs would be related to the reuse and maintenance process, and to encouraging the involvement of people. The six main critical factors identified in the current study (see Table 11.12) are related to reuse and maintenance process (Intr-C, Intr-D, and Intr-F) and to encouraging people to use SRPs using different artefacts (Intr-A, Intr-B, and Intr-E). These results slightly differ with the critical factors for RPs identified in the survey-based empirical study (Subsection 2.3.6), where the most importance was given to the existence of a reuse method and tool support. Although the existence of tool support matches one of the critical factors of the current study, that is not the case for the existence of a reuse method (only 3 interviewees stated that critical factor). The reason behind that is probably that the main alternative to use SRPs was explained during the presentation about SRPs and the PABRE framework done during the interviews, therefore the participants did not consider having a method as a critical factor, since a method was already incorporated in the proposal.

Cons from using SRPs and the barriers for their successful adoption would be related to the reuse and maintenance process, to the people involved, and to the impact on the final product. As shown in Table 11.11 and Table 11.13, aspects related to the reuse and maintenance process of SRPs correspond to the most frequent cons and barriers (Con-E, Con-F, Con-G, Con-H, Bar-C, and Bar-E). In addition, two of the most common cons and barriers were related to the people involved in the use of SRPs (Con-D and Bar-D). It is not surprising that they were considered important since, as stated by Dybå [145], in organizational processes, the involvement of personnel is a key factor for the adoption of a reuse technique and its success. Finally, another downside of the SRP approach identified in the study is the impact on product innovation that SRPs could have (Con-F).

The two most important barriers identified in the survey-based empirical study directly match the most frequent barriers identified in the current study (i.e. the resistance to change of requirements engineers and the integration of the patterns catalogue with the existing RE processes) (Subsection 2.3.6).

11.6 Threats to Validity

In this section, the validity threats of the study are discussed. As suggested by Wohlin et al. [108], they are presented in terms of construct, conclusion, internal and external validity.

Construct Validity

Construct validity concerns the relationship between observations from the study and the construct behind the research. To strengthen this aspect, this study was supported by two main principles: rigorous planning of the study, and the establishment of protocols for data collection and data analysis as suggested by Runeson and Höst [212]. Additionally, the instrument used to gather data (i.e. the interview guide) was carefully designed and piloted with 2 academic people that had an extensive background in industry (these interviews were discarded for the real study). The interview guide was designed in English, and the interviews were held in English too (except in two cases were the interview was carried out in Spanish). Although English was not the mother tongue neither of the interviewees nor of the interviewer, no problems of communication nor of understanding were experienced. Furthermore, the pilots helped to improve the understandability of the questions with respect to the use of suitable vocabulary that the participants were familiar with. However, there existed terminology differences between the different interviews. This was addressed by: a) asking clarifications questions during the interviews when needed, and b) applying multiple codes to the same statement to capture multiple interpretations. Finally, both in the interview guide and during the actual interview, the participants were aware that the data they provided would be confidential, anonymised, and aggregated with the rest of interviews, so the respondents could freely share their real experiences and perceptions.

Conclusion Validity

Conclusion validity is concerned with the ability to draw correct conclusions from the study. This threat was addressed from different perspectives.

The concepts (i.e. codes) were generated according to the process explained in Subsection 11.2.7. Throughout the coding, many concepts and their relationships were identified. However, as the research process continued, the concepts were merged and updated to develop the final categories. Traceability from the raw data to the categories and their relationships was preserved. Different types of triangulation were used to minimize possible biases. Different coding techniques (theory triangulation) were used to capture various aspects of a phenomenon. Selected cases from the dataset were analysed by two of the conductors to identify and to eliminate individual biases (researcher triangulation).

Internal Validity

Internal validity threats are related to factors that affect the causal relationship between the treatment and the outcome.

With respect to the data gathering strategy, relevant decisions were taken for achieving a further understanding of RE industrial practices. One of the main relevant decisions was to focus most of the questions of the interview guide on a single software development project. In this way, it was possible to further inquire and analyse specific contexts that generated a particular decision. This enhanced the value of the analysis and observations, as it allowed the understanding of the rationale behind certain RE practices and opinions. Nevertheless, some possible biases may be related to this strategy, for instance the fact that some time had passed since the project was completed, so it could be difficult for the participants to remember some project details. To reduce the possible side effects of this, the interview guide was sent in advance to the respondents so that they were informed of the kind of questions they were going to be asked and they could chose the project in advance and could fill the details of section A of the interview guide (personal, organization and project information). As a result, when performing the interviews, the participants rarely had difficulties remembering project details. Another factor that might affect the results was that the participants could have selected the most successful projects. To minimize this risk, it was explained to them that the study was not focused on analysing “wrong practices” but on knowing “how it is done in industrial practices”. In addition, all the respondents were open to follow-up contacts if clarifications on their answers was needed while analysing the data.

With respect to the data analysis strategy, all the interviews were recorded. The notes taken by the interviewer during and after the interview were completed with all the details provided by the participants by listening to the audio as much times as needed. This contributed to a better understanding and assessment of the data gathered. Not having transcribed the whole interviews could have represented a threat. However, to mitigate the issue, the audio was imported to the qualitative data analysis tool used (i.e. Atlas.ti), which offered the same coding functionalities both in audio files and text files (which would be the format used for the transcript of the interviews). This functionality was used to code the interviews' audios with *initial* and *structural* coding strategies. This allowed to access easily the audio related to a specific question when doubts on the improved notes arose during analysis.

To address a single researcher bias in the coding process, triangulation was applied. Selected interviews were analysed independently by two researchers and the results were discussed to identify and eliminate any individual biases. Responses were triangulated too, especially in the case of respondents from the same organization, in order to strengthen the correct understanding of the results. In addition, the generated categories were analysed, discussed and reviewed by all researchers of the team to ensure their accuracy, understanding and agreement. Furthermore, categories were checked with respect to the data gathered in order to confirm that none of the stated categories refuted any of the conclusions.

External Validity

External validity refers to the ability to generalize the results of research efforts to industrial practice. It is important to highlight that qualitative studies, such as the one presented in this chapter, rarely attempt to make universal generalizations beyond the studied setting. Instead, as Robson explains [211], they are more concerned with characterizing, explaining and understanding the phenomena under the contexts of study. To strengthen the external validity, several aspects were addressed. Some of the most relevant ones are listed.

First, the organizations in this study were selected by a strategy combining convenience sampling [211] and maximum variation sampling. The use of this convenience sampling approach reflects the difficulty in gaining industrial participation in these kinds of surveys. Any possible bias traditionally related to convenience sampling was tried to be mitigated by combining a maximum variation sampling, so that the approached organizations covered different characteristics regarding size, application domain, and business area.

Second, another aspect strengthening the external validity was that the interviewees were completely free for selecting a project for the interview, and the conductors of the study had no influence over this decision.

Third, the approached projects were of different size and types, and the interviewees had different backgrounds (see Section 11.3). Nevertheless, most of the resulting sampling organizations were developing web applications, embedded systems, or systems focused on the telecommunication domain. It is possible that this factor may have an impact on how the participants saw the viability of SRPs. Therefore, it is important to highlight that the findings of this study might be considered more relevant for this type of organizations or systems.

Finally, it is worth pointing out that, in the questions related to ES-RQ, the respondents were giving only their opinion as RE experts, so it cannot be assumed, for instance, that every benefit and drawback that was marked as relevant by the participants would actually appear when using SRPs. The study's findings should not be taken as assertions but as potential hypotheses that need to be further validated.

11.7 Conclusions

Through this chapter, it has been presented a preliminary analysis of the results of an interview-based qualitative study which main goal was to investigate the perception of participants about the usefulness, benefits and drawbacks of the PABRE framework. The analysis took into consideration 22 interviews of IT professionals with experience as requirements engineers in industry from 11 Swedish organizations.

The main findings of the study reveal that SRPs are considered a good technique to achieve requirements reuse in different types of organizations and projects. The study also shows that SRPs are preferred to be used during the elicitation and specification of requirements, either searching, selecting and applying SRPs, or using the SRP catalogue to check that no requirement has been forgotten. Both of these approaches of use are considered in the PABRE framework (as shown in Section 5.2). In addition, the interviews show that SRPs are not relevant for all types of requirements, but for the ones that are recurrent among projects (which usually are NFRs and some FRs). Apart from the SRP catalogues for NFRs and recurrent FRs for the CMS domain, the PABRE framework incorporates also a catalogue for NTRs. The decision to create SRPs for NTRs is supported by the fact this kind of requirement is in fact quite recurrent, but usually only in call-for-tender processes.

The benefits identified in the study by the use of SRPs are mainly related to the time spent for the RE process and the quality of the resultant SRSs, which corroborate the possible benefits of the PABRE framework presented in Section 3.4. Specifically, the participants of the interviews mentioned the fact that SRPs could help to have less unambiguous and more uniform requirements, and more complete SRSs. The identified critical factors for the successful adoption of SRPs would be related to the reuse and maintenance process, and to encouraging the involvement of people. With the aim to ease and motivate this introduction process, the PABRE framework incorporates some artefacts that pursue that purpose. Concretely, it incorporates finished catalogues of SRPs, a ROI model and tool support.

Regarding the cons identified from using SRPs and the barriers identified for their successful adoption, they are related to the reuse and maintenance process, to the people involved, and to the impact on the final product. Table 11.14 shows the relationships among the possible drawbacks of the PABRE framework presented in Section 3.5 and the cons and barriers identified in this study. As can be seen, almost all the drawbacks are corroborated by the interviews' answers. However, only two of them are strongly supported by the interviewees (those being *The integration of the requirements reuse method* and *The cost of maintenance of the SRP catalogue*). Therefore, the rest are less obvious and further validation would be needed to prove their appearance when actually using SRPs in organizations.

It is important to remark that the two cons identified in the interviews in relation to the creation of the SRP catalogue and its completeness (Con-E and Con-G) have not been considered as actual drawbacks of the PABRE framework, because they could be easily addressed: getting the right SRP catalogue (Con-E) can be easily overcome by using the method to create SRPs using SRSs of the organization that will use the catalogue, and not trusting on the completeness of the catalogue (Con-G) could be easily overcome by teaching to the SRP users what they can expect to find in the catalogue. Probably, participants stated those drawbacks because during the presentation of the framework done during the interviews, it was not possible to explain all the details of the proposal.

As a general conclusion, this study gives a positive answer to its main research question: *Do SRPs, as defined in the PABRE framework, facilitate requirements reuse?*, corresponding to ES-RQ of the study.

Table 11.14 – Mapping among the possible drawbacks of the PABRE framework and the cons and barriers identified in the interviews

Drawback [1]	Related Code and Code Name [2]	Frequency [2]	Percentage [2]
The integration of the requirements reuse method	*Bar-C*: Integration with organization's processes	14	64%
	Bar-D: Resistance to change	13	59%
The cost of maintenance of the SRP catalogue	*Con-G*: Maintainability of SRPs	5	23%
	Bar-E: Maintenance	7	32%
The "heaviness" of the reuse process	*Con-K*: More time needed for the RE stage	1	5%
	Con-L: Learning curve	1	5%
	Bar-H: Heavier RE stage	2	9%
The "inefficiency" of the reuse process	*Con-K*: More time needed for the RE stage	1	5%
	Con-L: Learning curve	1	5%
	Bar-H: Heavier RE stage	2	9%
The context of use of an SRP catalogue	---	---	---
SRP impact in innovation	*Con-F*: Less creative products	6	27%
SRP impact on requirements	*Con-D*: Get too used	7	32%

[1] Explanation in Section 3.5

[2] Definition, frequency and percentage of the categories extracted from Table 11.11 or Table 11.13

Conclusions and Future Work

CHAPTER 12 Conclusions and Future Work

12.1 Conclusions

The research questions of this thesis are stated in Chapter 1. In this section, answers to these questions are provided.

> **RQ1** *In the field of RE, do pattern-based approaches proposed so far provide an adequate technique to achieve RE knowledge reuse during the elicitation and specification of requirements?*

Answer. The SMAP carried out in this thesis (Chapter 2) has shown that there is neither any widespread approach used in organizations, nor any complete approach that incorporates all the necessary elements to encourage organizations to adopt requirements reuse. Although there are multiple approaches for reusing requirements through patterns during the elicitation and specification of requirements, most of these approaches are fuzzy, in the sense that they define precisely the pattern approach but they do not take into account other assets that are needed to integrate the approach in real environments. Specifically, the critical points not covered by more than half of the approaches of the SMAP are: establishing relationships among the artefacts to reuse; arranging and classifying them to ease their access; incorporating a metamodel of the artefacts to reuse and their arrangement; proposing methods to construct and update the artefacts; defining a catalogue of artefacts general enough and finished so it can be used as a base example to construct other catalogues; implementing tools to facilitate the incorporation of the approach in real environments; and testing the approaches in real environments, proving its validity and the economic benefits that the use of the approach may imply.

In addition, the survey-based empirical study presented also in Chapter 2 about the state of the practice of requirements reuse and patterns, concludes that requirements reuse is not an established practice in IT organizations and that most of the participants reusing requirements is doing it in ad-hoc manner, with a very small percentage of them using RPs. The empirical study also identifies that the two most critical aspects for the successful adoption of RPs are the existence of a reuse method specific for the approach as well as having a tool that supports it. Finally, the empirical study also shows the general believe that RPs could mitigate problems related to the quality of SRSs and, in the case of participants using some kind of RP, it is believed that RPs could help to spend less time for the elicitation and specification of requirements.

> ***RQ2*** *What is the best structure and semantics that SRPs should have to be applied over FRs, NFRs, NTRs)?*

Answer. The answer to this question has been the design of the metamodel that defines the structure and semantics of SRPs and their organization inside a catalogue (Chapter 4). The metamodel has been extracted from NFRs stated in real SRSs, but also from literature review and experience knowledge. Afterwards, it has been validated with FRs and NTRs (Chapter 10). The results of this validation entailed small changes in the SRP structure, which have been already incorporated in the metamodel. In addition, the quality of the metamodel has been checked with regard to two metamodel quality frameworks, achieving positive results in both cases.

> ***RQ3*** *How can SRPs be integrated into the RE activities and processes so that their application yields benefits that justify the cost of their adoption?*

Answer. The approach to face this RQ has been the design of different methods to use, manage and update SRPs (Chapters 5 and 6), and the implementation of a software system that supports these methods (Chapter 9). In order to ease the introduction of SRPs in organizations, a complete and finished SRP catalogue for FRs, NFRs and NTRs has been constructed (Chapter 7), which can be used as a base to construct more SRPs in organizations. Furthermore, a ROI economic model has been designed, with the goal of calculating the economic benefits of the approach (Chapter 8). The methods to use SRPs and the types of requirements incorporated in the catalogue align with the results of the empirical study based on semi-structured interviews (Chapter 11). In addition, the approach presented in this thesis covers the critical factors for successfully adopting an SRP catalogue, identified in the same study, which were related to the reuse and maintenance process, and to the encouragement of the involved people using SRPs. The participants of this study also agreed that SRPs could help to spend less time during the elicitation and specification of requirements and to achieve higher-quality SRSs, which align with the possible benefits presented in Section 3.4. Specifically, the individual requirements elicited and specified using SRPs will be unambiguous, consistent and singular, and more complete, feasible, traceable and verifiable. Regarding SRSs as a whole, they will be consistent and more complete. The participants of the above-mentioned empirical study specifically mentioned that SRPs could help to have unambiguous and uniform requirements, and more complete SRSs.

As a final conclusion, the most valuable outcome of this thesis as a whole is the design and implementation of the PABRE framework (Chapter 3), which joins the results of RQ2 and RQ3, to manage SRPs, and use them during requirement elicitation and specification processes of IT projects of any software domain. SRPs capitalise requirements knowledge that is convenient to reuse from previous projects (specifically, this knowledge corresponds to textual requirements). The aim of the framework is improving the quality and validity of SRSs and reducing the time spent during requirements elicitation and specification processes, with the economic cost savings that both entail.

The framework includes all the necessary elements to encourage organizations to adopt it, and covers almost all the critical points detected in the SMAP and in the empirical studies based on surveys and interviews. Specifically, the framework embraces: 1) A metamodel that describes the structure and semantics of SRPs, the relationships among them, and their organization into a catalogue; 2) Methods for guiding the use of an SRP catalogue during requirements elicitation and specification, as well as other ones for constructing and updating it; 3) A finished SRP catalogue that incorporates SRPs for FRs, NFRs and NTRs, the functional ones being specific for the CMS domain; 4) An economic model to perform cost-benefit analysis on the adoption of SRPs based on ROI; and 5) The PABRE system as technological support.

The only point that still remains to be done in the PABRE approach is the validation in real environments. However, the proposed economic model allows to calculate the ROI of the requirements reuse approach, and the academic validation done in the interviews-based empirical study points out that the PABRE framework could drive to higher-quality SRSs elicited and specified in less time, thus reducing the economic cost of IT projects.

12.2 Comparison with Other Approaches

Due to the application of scope of the PABRE framework, it is difficult to identify a single approach to which the framework can be compared. To make the comparison easier, the same aspects evaluated in the SMAP presented in Section 2.2 have been analysed. The proposals compared in this section correspond to the ones identified as being the most consolidated ones in the SMAP. In addition, Withall's proposal (P77) has also been considered in this analysis, as it is one of the most important references when talking about requirement patterns. As can be seen in Table 12.1, Table 12.2 and Table 12.3, the PABRE approach is the most consolidated proposal according to the criteria used to identify the most consolidated proposals in the SMAP (the criteria are explicitly stated in Subsection 2.2.3.3). The next paragraphs summarize the relevant facts resulting from the comparison analysis.

The simplest version of patterns is proposed in Monzon's et al. work (P45) and Panis work (P50), which focus on reusing the entire content of SRSs (i.e. FRs, NFRs and NTRs) to create new SRSs, being the smallest entity to reuse an individual requirement. One of the critical points of these approaches is that the reused requirements (expressed as SNL) are not accompanied by any other information, such as the conditions when a requirement could be reused. Monzon et al. also sketches a particular reuse methodology that allows different types of requirements: strong reused requirements, which are those ones considered to evolve synchronously in the future (i.e. if a requirement is changed the cloned requirements also change); weak reused requirements, which are those ones that are copied from a project source but evolve separately from the source; and new requirements for a particular project. These types are kind of similar to the ones used in the PABRE framework: requirements can be reused as they are, with small modifications or creating completely new requirements. Panis' work also incorporates a specific method for reusing patterns, a hierarchical organization of patterns (as PABRE does), and a catalogue of patterns (specific for a project). However, none of the approaches consider the update of patterns neither a metamodel for them. Unlike Panis (where no tool is presented), in Monzon's et al. proposal an implemented tool supports completely the reuse process. However, the proposal does not take care of the organization of requirements (i.e. possible relationships and classifications).

Table 12.1 – Comparison of the PABRE framework with other approaches (1/3)

	Proposal ID	*PABRE*	P16	P41	P45	P49	P50	P61	P66	P77
General Aspects	Studies ID	*[81], [86], [76], [94], [96], [87], [77], [78], [82], [84], [85], [73], [79], [85]*	[SMAP-18]	[SMAP-48]	[SMAP-54]	[SMAP-58]	[SMAP-63]	[SMAP-79]	[SMAP-84]	[SMAP-101]
	Year of publication	*2010, 2010, 2011, 2011, 2011, 2012, 2013, 2013, 2013, 2013, 2013, 2014, 2014, 2014*	2013	2012	2008	2015	2015	2007	2013	2007
	Type of venue	*Conf., Report, Report, Conf. Work., Work., Report, Work., Book section, Conf., Journal, Work., Conf., Conf.*	Journal	Conf.	Conf.	Journal	Conf.	Conf.	Journal	Book
	Author(s) countries	*Luxembourg, Mexico, Spain*	Germany	Spain, USA	Spain	Mexico, Spain	USA	Korea	India	USA
Overview	Classification of the proposal	*Patterns with context information*	Patterns with context information	Patterns with context information	Templates	Patterns with context information	Templates	Patterns with context information	Patterns with context information	Patterns with context information
	Scope	*Elicitation, Specification*	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	*General purpose*	Embedded systems	General purpose	General purpose	General purpose	General purpose	Embedded systems	Online exami-nation systs.	General purpose
	Domain generalizable?	*---*	Yes	---	---	---	---	Yes	Yes	---
	Type of reqs	*F, NF, NT*	F, NF, NT	F, NF	F, NF, NT	F, NF, NT	F, NF, NT	F, NF, NT	F, NF, NT	NF, NT
	Artefact to reuse	*SNL*	SNL, Feature models	SNL, Business process models	SNL	SNL	SNL	SNL	SNL	SNL
	Notation	*NL*	NL, Variability models	NL, A own one	NL	NL	NL	NL	NL	NL

Table 12.2 – Comparison of the PABRE framework with other approaches (2/3)

	Proposal ID	*PABRE*	P16	P41	P45	P49	P50	P61	P66	P77
Structure	Structure	***BS***	BS	BS	IK	BS	IK	BS	BS	BS
	Structure information	***parameterized requirements sentences + metadata + applicability conditions***	requirements + metadata + relation to features in feature models + other stages information	requirements + metadata + diagram + other stages information	requirement	parameterized requirements sentences+ metadata + applicability conditions	requirement	Freqs. Sentences + quality reqs. + metadata + applicability conditions + other stages information	requirement sentence + metadata + applicability conditions	parameterized requirements + applicability conditions + other stages information
Organization	Relationships	***Yes (conflicts, disjoint, relates-to, requires, rule, subset, supports)***	Yes (traceability relationships, no types specified)	Yes (shown in a diagram inside the pattern, no types specified)	No	No	No	Yes (field *Related Features*, no types specified)	Yes (fields *Extra Require-ments* and *Related patterns*, no types specified)	Yes (*has*, *displays*, *uses*, *is-a*)
	Arrangement	***RPS***	RPS	NS	NS	TX	TX	RPS	RPS	RPS
	Classification	***Yes***	No	NS	NS	Yes	Yes	Yes	Yes	Yes
Metamodel	Reuse artefact	***Yes***	No	No	No	No	No	Yes	No	No
	Arrangement	***Yes***	No	No	No	No	No	Yes	No	No
Methods	Construction	***Yes***	Yes	No	Partial	Yes	Yes	Yes	Partial	Yes
	Reuse	***Yes***	Partial	No	Partial	Yes	Yes	Yes	Yes	Yes
	Update	***Yes***	No	No	No	Yes	No	Partial	No	Partial
Catalogue	Exists?	***Yes***	Yes	No	Yes	No	Yes	Yes	Yes	Yes
	Domain	***General***	Project	---	Project	---	Project	Project	Project	General
	State	***Finished***	NS	---	NS	---	NS	NS	Finished	NS

Table 12.3 – Comparison of the PABRE framework with other approaches (3/3)

	Proposal ID	*PABRE*	P16	P41	P45	P49	P50	P61	P66	P77
Tools & Validation	Related tools	***Managing patterns (CRUD), managing classification schemas (CRUD), creating SRSs using patterns and managing them (CRUD).***	IBM Rational Doors tool for managing requirement patterns, and Pure Variants for managing the variability models.	No	Reusing reqs from previous SRSs: the tool allows selecting for each new SRS what are the reqs that can be reused in other projects, as well as creating new SRSs creating 3 types of reqs.	Requisite Pro plugin for: managing user roles, creating the catalogue, searching in the catalogue, select reusable reqs, validate reusable reqs, adapt reusable reqs, and managing the catalogue.	No	No	No	No
	Tests	***No***	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	Type of research	***Empirical***	Experience report	Non-empirical	Empirical	Empirical	Experience report	Empirical	Non-empirical	Non-empirical

Conf. –> Conference; *F* → Functional; *NF* → Non-Funcitonal; *NT* → Non-Technical; *SNL* → Sentences in Natural Language; *NL* → Natural Language; *BS* → Bigger Structure; *IK* → Isolated Knowledge; *RPS* → Repository; *TX* –> → Taxonomy; *NS* → Not Stated

Pacheco's et al. work (P49) and Srivastava's work (P66) define the requirement pattern concept as a set composed of SNL, some metadata fields (that vary depending on the approach) and the conditions that state when the pattern is applicable. In the case of Srivastava, the SNL also incorporate parameters to adapt them to a specific project, and the relationships with other patterns are considered in each pattern. Regarding the metadata, both approaches propose a rich set of metadata features. Some of them are similar to the ones identified in PABRE, such as *applicability* (similar to the *description* in PABRE), *author*, *source*, *goal*, and *constraints* (similar to the *behaviour* in PABRE). Others are quite different from the ones of PABRE, but might give good ideas for future improvements of the SRPs (e.g. *importance*, *criticality* and *viability*). Pacheco's et al. approach highlights in the definition of methods to construct, use and update their patterns, but no metamodel for patterns neither a catalogue of them is presented. In contrast, Srivastava's work only defines a method to use patterns, sketches some guidelines to build them, and presents a catalogue that, although specific for an example project, is finished. However, the update of patterns is not considered, neither a metamodel for them. Additionally, Pacheco's et al. work also incorporates a tool (implement as a plug-in of the commercial tool Requisite Pro) to use and manage requirement patterns, and even some tests of the approach have been carried out.

The PABRE framework and Withall's proposal (P77) define requirement patterns in a similar way. There, requirement patterns are understood as one or more parameterized SNL, where each one corresponds to a requirement, some metadata fields, and the applicability conditions. Both approaches distinct alternative ways to achieve the same goal (for instance, the same requirement can have more or less detailed information, or it could be the case that for different contexts the requirement is expressed in different ways), consider relationships among patterns and organize the requirements in a repository. The main differences between these approaches are:

- The PABRE framework covers FRs, NFRs and NTRs, while Withall's patterns are focused on NFRs and NTRs.
- Withall's work incorporates information for further development stages, such as design.
- A metamodel for both the requirement patterns and their classification is only incorporated in the case of PABRE.

- Withall defines in his work a methodology to construct and use patterns and, although in an informal manner, how to update the defined catalogue. These methods are fully detailed in the case of the PABRE framework.
- Tools to ease the use of patterns are only taken into account by the PABRE approach.
- Although none of the proposals have been tested in real projects, the PABRE framework is the only one that has been validated (even if it is only from an academic point of view).

Finally, Macasaet's et al. proposal (P41), Derakhshanmanesh's et al. proposal (P16) and Lee's et al. proposal (P61) go one step further and, apart from incorporating SNL and metadata, incorporate other information to reuse. Macasaet et al. takes into account diagrams expressed in an own language (similar to a business process modelling notation) in order to show what is the global view of the pattern; Derakhshanmanesh et al. incorporates the use of feature models, family models and variant description models, to reuse requirements in software product lines; and Lee et al. groups FRs with the NFRs that may have an effect on the FRs. The requirements in the patterns are, therefore, accompanied by much more information than in the case of the PABRE framework. As a difference from the PABRE SRPs, Macasaet's et al. and Lee's et al. patterns also contain information that could be useful for other stages (e.g. the average time spent on coding the requirement). In addition, as in the PABRE framework, in Derakhshanmanesh's et al. work it is not mandatory to reuse all the requirements in the pattern: that is achieved in Derakhshanmanesh's et al. proposal by marking in the patterns if the requirements are mandatory or optional, and in the case of the PABRE framework with the use of fixed and extended parts. The three approaches consider the relationships among patterns, but not so much information is provided about these relationships, as the proposals just mention that relationships are incorporated as a feature in the patterns. However, a downside of the three patterns is that the SNL on them are not parameterised, and the method to reuse them is either not defined or it does not consider the possible modification of the sentences, which makes the reuse quite rigid. This aspect, though, is contemplated in the PABRE framework.

In the three approaches before-mentioned, a method to construct the catalogue is only incorporated in Derakhshanmanesh's et al. work and Lee's et al. work, while a method to update the catalogue is only considered in the last one. In addition, Derakhshanmanesh's et al. and Lee's et al. proposals consider the organization of their patterns into a repository (as PABRE does), but only the last one presents a specific classification for them. Regarding metamodels, only Lee's et al. approach defines a metamodel for both the requirement patterns and their organization in a catalogue. Another critical aspect of the proposals is the fact that the catalogue of patterns proposed is, in the case of Macasaet et al., inexistent (only some examples are provided), and in the case of Derakhshanmanesh et al. and Lee et al., it is specific for a project but there is no information about if the catalogue is finished yet. Regarding tools, the reuse functionalities have only been implemented in Derakhshanmanesh et al. by developing plugins of the commercial tools IBM Rational Doors and Pure Variants, which could make the integration in organizations easier. Finally, Derakhshanmanesh et al. corresponds to the most tested proposal, as the work corresponds to an experience report and there is available data about its use.

12.3 Future Work

Both the work in Part I (the PABRE framework) and Part II (its validation) can be extended in different directions.

In Part I, the PABRE metamodel could be accompanied by a model of quality for the SRP catalogue, which will incorporate metrics that characterize its quality. The PABRE method to create SRPs can be extended to use natural language processing techniques to detect ambiguities in the SRP catalogue constructed. A more ambitious approach to the proposed method would be the semi-automatic creation of patterns, again by using natural language processing techniques over the SRSs to detect the most recurrent requirements, which will be the base to construct SRPs. In the same way, the PABRE method to update the SRP catalogue could be semi-automatized, by proposing automatically the changes that are necessary to do in the catalogue; the changes would be based on rules (e.g. "a new part in an SRP cluster is needed if there are more than X requirements related to this cluster in Y different IT projects") and the RE expert would be the one in charge of evaluating the real need of the change and of implementing it. Another interesting aspect to be considered in both the creation and update methods of the SRP catalogue is the possibility of using natural language processing techniques to detect duplicates or similar

SRPs that could be merged. The PABRE method to use SRPs to elicit and specify requirements could be extended by adding an SRP recommendation system that introduces in the method recommendation capabilities such as "projects that used this pattern usually use this other".

Additionally in Part I, the PABRE ROI economic model could be extended by adding more software metrics (such as technical debt, degeneration over time, risk metrics, and indirect benefits). Finally, regarding the PABRE system, there are different extensions that could be tackled: 1) Developing new functionalities in PABRE-Proj, specially focused on expanding the search of SRPs and on adding the SRP recommendation system; 2) Expanding the functionality of exporting SRSs to other requirement management tools, by adding more exportation formats (e.g. to IBM Rational Doors[39], JAMA[40], and Visure Requirements[41]); 3) Integrating into the PABRE system the prototype that automatically proposes SRPs while writing down requirements (which are not elicited using the catalogue), as well improving and expanding the prototype functionalities.

In Part II, the validation can be extended with evidence of the correctness, suitability and benefits of the PABRE framework in real environments, by testing and evaluating the use of SRPs in elicitation and specification processes of real organizations. This will be achieved through a case study. In addition, in order to validate the ROI economic model, the case study shall encompass different IT projects of the same organization, which will elicit and specify requirements using the PABRE framework, thus being able to check the appropriateness of the economic model with the data gathered in these projects. However, before doing so, it will be necessary to validate the obtained results through selection experiments, trying to solve possible issues that would appear while doing the case study, so the results of this last one will not be deviated by these issues. Regarding the empirical studies carried out, it is planned to expand the conclusions of the studies by proposing a maturity model for reuse in organizations based on the studies' results. In this model, practices, techniques, tools and actors that are more suitable for each maturity level will be defined and described in order to help organizations introduce or improve requirements reuse.

[39] http://www-03.ibm.com/software/products/en/ratidoorfami
[40] http://www.jamasoftware.com/
[41] http://www.visuresolutions.com/

List of Abbreviations, Appendixes and Bibliographies

List of Abbreviations

API	Application Programming Interface
CIO	Chief Information Officer
CMS	Content Management System
CL	SRP Cluster
EBSE	Empirical-Based Software Engineering
FP	Fixed Part
F	Functional
EP	Extended Part
FR	Functional Requirement
IT	Information Technology
NF	Non-Functional
NFR	Non-Functional Requirement
NT	Non-Technical
NTR	Non-Technical Requirement
PABRE	PAttern-Based Requirements Elicitation
RE	Requirements Engineering
RMT	Requirements Management Tool
ROI	Return-On-Investment
RP	Requirement Pattern
SME	Small and Medium Sized Enterprise
SMAP	Systematic Mapping
SNL	Sentences in Natural Language

SRP	Software Requirement Pattern
SRS	Software Requirement Specification

Appendix I – SMAP Need Identification Works

This appendix contains the papers analysed during *The Identification of the Need for a Study* stage (see Subsection 2.2.2.1) of the SMAP presented in Section 2.2.

V. Alves, N. Niu, C. Alves, and G. Valença, "Requirements engineering for software product lines: A systematic literature review", in Information and Software Technology, 2010.

F.B. de Azambuja, R. M. Bastos, and A. P. Terra Bacelo. "Systematic Review of Requirements Reuse", in International Conference on Software Engineering and Knowledge Engineering (SEKE), 2009.

M. Babar, M. Imran, M. Ghazali, and D. NA Jawawi, "Systematic reviews in requirements engineering: A systematic review", in IEEE Malaysian Software Engineering Conference (MySEC), 2014.

N. Bakar, N. Hasrina, Z. M. Kasirun, and N. Salleh, "Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review", in Journal of Systems and Software, 2015.

M. Bano, D. Zowghi, and N. Ikram, "Systematic reviews in requirements engineering: A tertiary study", in IEEE Fourth International Workshop on Empirical Requirements Engineering (EmpiRE), 2014.

X. Franch, "Software requirement patterns: a state of the art and the practice", in IEEE International Conference on Software Engineering (ICSE), 2015.

I. Iankoulova, and M. Daneva, "Cloud computing security requirements: A systematic review", in IEEE International Conference on Research Challenges in Information Science (RCIS), 2012.

A. F. Jaramillo, "Requirements elicitation approaches: A systematic review", in IEEE International Conference on Research Challenges in Information Science (RCIS), 2015.

D. Mellado, C. Blanco, L. E. Sánchez, and E. Fernández-Medina, "A systematic review of security requirements engineering", in Computer Standards and Interfaces, 2010.

H. Meth, M. Brhel, and A. Maedche, "The state of the art automated requirements elicitation", in Information and Software Technology, 2013.

J. G. Mohebzada, G. Ruhe, and A. Eberlein, "Systematic mapping of recommendation systems for requirements engineering", in International Conference on Software and System Process (ICSSP), 2012.

C. Palomares, X. Franch, and C. Quer, "Requirements reuse and patterns: a survey", in International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2014.

S. Sepúlveda, A. Cravero, and C. Cachero, "Requirements modeling languages for software product lines: A systematic literature review", in Information and Software Technology, 2016.

R. C. da Silva, and F. B. Vavassori Benitti, "Padrões de Escrita de Requisitos: um mapeamento sistemático da literatura", in Workshop on Requirements Engineering (WER), 2011.

A. Souag, R. Mazo, C. Salinesi, and I. Comyn-Wattiau, "Reusable knowledge in security requirements engineering: a systematic mapping study", in Requirements Engineering, 2015.

Appendix II – SMAP Data Extracted

This appendix contains the data extraction form (see Subsection 2.2.3.2) filled for each one of the 80 proposals of the SMAP presented in Section 2.2. In the following, the list of abbreviations used in the tables is provided:

- BS: Bigger Structure
- BPM: Business Process Models
- CM: Conceptual Model
- F: Functional
- IK: Isolated Knowledge
- NF: Non-Functional
- NS: Not Stated
- NT: Non-Technical
- Reqs: Requirements
- Rels: Relationships
- RPS: Repository
- SNL: Sentences in Natural Language
- SRS: Software Requirement Specification
- TX: Taxonomy
- UC: Use Case

	Proposal ID	P01	P02	P03
General Aspects	Studies ID	[SMAP-1]	[SMAP-2], [SMAP-3], [SMAP-97]	[SMAP-4]
	Year of publication	2014	2012, 2013, 2005	2012
	Type of venue	Conference	Workshop, Workshop, Conference	Conference
	Author(s) countries	Germany	Portugal	Germany
Overview	Classification of the proposal	Process patterns, Templates	Writing guidelines patterns, Formalization/Specification patterns	Templates, Formalization/Specification patterns
	Scope	Elicitation, Specification	Specification	Elicitation, Specification
	Domain	General purpose (S)	General purpose (S)	Security requirements
	Domain generalizable?	No	No	No
	Type of reqs	NS	F, NF, NT	F, NF, NT
	Artefact to reuse	RE practices, Can vary from template to template	Grammars	SNL, Formalization of requirements
	Notation	Natural language, Can vary from template to template	Natural language, Grammar language (NS)	Natural language, Formal Language (NS)
Structure	Structure	BS	IK	BS
	Structure information	requirement + metadata + applicability conditions	SNL structure	parameterized requirements sentences + formalization
Organi-zation	Relationships	No	No	No
	Arrangement	NS	NS	NS
	Classification	NS	NS	NS
Meta-model	Reuse artefact	No	Yes	No
	Arrangement	No	No	No
Me-thods	Construction	No	Partial	No
	Reuse	Yes	No	Yes
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	General	General	General
	State	NS	Evolution	NS
Tools & Validation	Related tools	No	Converting NL requirements into a formal requirements language by using the patterns proposed.	No
	Tests	No	No	No
	Type of research	Non-empirical	Non-empirical	Non-empirical

	Proposal ID	P04	P05	P06
General Aspects	Studies ID	[SMAP-5]	[SMAP-6]	[SMAP-7]
	Year of publication	2013	2014	2013
	Type of venue	Journal	Conference	Workshop
	Author(s) countries	Brazil	Spain	India
Overview	Classification of the proposal	Patterns with context information	Formalization/Specification patterns	Writing guidelines patterns
	Scope	Elicitation, Specification	Specification	Specification
	Domain	General purpose (NS)	General purpose (NS)	General purpose (S)
	Domain generalizable?	No	No	No
	Type of reqs	F, NF, NT	F, NF, NT	F, NF, NT
	Artefact to reuse	SNL	UC	SNL
	Notation	Natural language	BPMN, UML	Natural language
Structure	Structure	BS	BS	IK
	Structure information	requirement sentence + metadata + applicability conditions	use cases + metadata + BPMN + UC diagram + applicability conditions	SNL structure
Organi-zation	Relationships	No	No	No
	Arrangement	RPS	NS	NS
	Classification	Yes	NS	NS
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	No	Yes
	Reuse	Yes	No	No
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	No
	Domain	General	General	---
	State	NS	NS	---
Tools & Validatio	Related tools	No	No	No
	Tests	Yes	No	No
	Type of research	Empirical	Empirical	Non-empirical

	Proposal ID	P07	P08	P09
General Aspects	Studies ID	[SMAP-8]	[SMAP-9]	[SMAP-10]
	Year of publication	2013	2015	2006
	Type of venue	Book Section	Workshop	Conference
	Author(s) countries	Spain, Germany	Brazil	USA
Overview	Classification of the proposal	Patterns with context information	Domain/NF patterns	Domain/NF patterns
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	General purpose (NS)	Provenance requirements	General purpose (S)
	Domain generalizable?	No	No	No
	Type of reqs	F, NF, NT	NF	F, NF
	Artefact to reuse	SNL	Refinement models (in *i** to achieve NFRs)	UC
	Notation	Natural language	*i**	UML
Structure	Structure	BS	BS	BS
	Structure information	parameterized requirements sentences + + applicability conditions + other stages information	*i** model + metadata + applicability conditions	use case + associated requirements (F, NF) + metadata + applicability conditions + other stages information
Organi-zation	Relationships	Yes (parent-child, requires, relatedTo, mutuallyExclusive, reifies, discussionThread)	Yes (specialization-of, part-of, ocurrence-of)	Yes (generalization, specialization)
	Arrangement	RPS	NS	NS
	Classification	No	NS	NS
Meta-model	Reuse artefact	No	No	Yes
	Arrangement	No	No	No
Me-thods	Construction	No	No	No
	Reuse	Yes	Partial	No
	Update	No	No	No
Cata-logue	Exists?	No	Yes	Yes
	Domain	---	General	Project
	State	---	NS	NS
Tools & Validation	Related tools	Reusing requirements from a repository of patterns, as well checking that the requirements for a project are consistent according to the dependencies introduced in the patterns.	No	No
	Tests	No	No	No
	Type of research	Empirical	Non-empirical	Empirical

	Proposal ID	P10	P11	P12
General Aspects	Studies ID	[SMAP-11]	[SMAP-12]	[SMAP-13]
	Year of publication	2014	2012	2012
	Type of venue	Workshop	Workshop	Workshop
	Author(s) countries	Brazil	Nigeria, Norway, Austria	Nigeria, Norway
Overview	Classification of the proposal	Domain/NF patterns	Domain/NF patterns --> Ontologies, Templates	Templates
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	Awareness requirements	Security requirements	Security requirements
	Domain generalizable?	No	No	No
	Type of reqs	NF	F, NF, NT	F, NF
	Artefact to reuse	Refinement models (in *i**), Problem frames	SNL, Ontologies	SNL
	Notation	*i**, Natural language	Natural language, Ontology language (not stated)	Natural language
Structure	Structure	BS	IK	IK
	Structure information	*i** models + applicability conditions	parameterized requirements + ontology	parameterized requirement sentence
Organi-zation	Relationships	No	No	No
	Arrangement	NS	NS	NS
	Classification	NS	NS	NS
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	No	No
	Reuse	Partial	Yes	Yes
	Update	No	No	Partial
Cata-logue	Exists?	Yes	Yes	No
	Domain	General	Project	---
	State	NS	NS	---
Tools & Validation	Related tools	No	Given a misuse case, the tool automatically analyses it, identify the threats using the ontology and propose boilerplates to specify new security requirements to countermeasure the threats.	Boilerplates are proposed to users automatically taking into account the misuse cases introduced by the user. Ontologies are used for helping users to instantiate the parameters in boilerplates.
	Tests	No	Yes	Yes
	Type of research	Non-empirical	Empirical	Empirical

	Proposal ID	P13	P14	P15
General Aspects	Studies ID	[SMAP-14]	[SMAP-15], [SMAP-16]	[SMAP-17]
	Year of publication	1996	2006, 2005	2003
	Type of venue	Conference	Conference, Workshop	Conference
	Author(s) countries	USA, United Kingdom, Brazil	USA	Germany, Canada
Overview	Classification of the proposal	Domain/NF patterns	Domain/NF patterns	Writing guidelines patterns
	Scope	Elicitation	Elicitation, Specification	Specification
	Domain	General purpose (S)	Agent-based systems	Embedded systems
	Domain generalizable?	No	No	No
	Type of reqs	F	F, NF	F, NF, NT
	Artefact to reuse	KAOS diagrams	SNL, UC	Grammars
	Notation	Temporal formal logic	Natural language	Natural language, Grammar language (NS)
Structure	Structure	BS	BS	BS
	Structure information	*i** model + metadata	requirement + applicability conditions	SNL structure + guidelines
Organi-zation	Relationships	Yes (weakening, strengthening)	No	No
	Arrangement	RPS	NS	TX
	Classification	No	NS	No
Meta-model	Reuse artefact	No	No	Yes
	Arrangement	No	No	No
Me-thods	Construction	Partial	Yes	No
	Reuse	Partial	Yes	No
	Update	Yes	Yes	No
Cata-logue	Exists?	Partial	Yes	No
	Domain	General	Project	General
	State	Evolution	NS	NS
Tools & Validatio	Related tools	No	No	No
	Tests	No	No	Yes
	Type of research	Non-empirical	Non-empirical	Empirical

	Proposal ID	P16	P17	P18
General Aspects	Studies ID	[SMAP-18]	[SMAP-19]	[SMAP-20]
	Year of publication	2013	2013	2014
	Type of venue	Journal	Workshop	Journal
	Author(s) countries	Germany	Canada	Ireland, Netherlands
Overview	Classification of the proposal	Patterns with context information	Process patterns	Formalization/Specification patterns
	Scope	Elicitation, Specification	Elicitation, Specification	Specification
	Domain	Embedded systems	General purpose (S)	Compliance requirements
	Domain generalizable?	Yes	No	No
	Type of reqs	F, NF, NT	---	NS
	Artefact to reuse	SNL, Feature models	RE practices	Formalization of requirements
	Notation	Natural language, Variability models language (NS)	Natural language	Natural language, Temporal formal logic, A own one
Structure	Structure	BS	BS	IK
	Structure information	requirements + metadata + relation to features in feature models + other stages information	best way to elicit/specify state machines for behavioural reqs + metadata + applicability conditions	formalization
Organization	Relationships	Yes (traceability relationships, no types specified)	No	Yes (no types specified)
	Arrangement	RPS	NS	TX
	Classification	No	NS	Yes
Meta-model	Reuse artefact	No	No	Yes
	Arrangement	No	No	No
Methods	Construction	Yes	No	No
	Reuse	Partial	No	No
	Update	No	No	No
Catalogue	Exists?	Yes	Yes	Yes
	Domain	Project	General	General
	State	NS	NS	NS
Tools & Validation	Related tools	IBM Rational Doors tool for managing requirement patterns, and Pure Variants for managing the variability models.	No	Three tools. The interesting one for the SMAP is the Compliance Rule Manager tool, for constructing reqs using patterns, and transforming the reqs into formal logic, so they can be checked by the other 2 tools against the business process models.
	Tests	Yes	Yes	Yes
	Type of research	Experience report	Empirical	Empirical

	Proposal ID	P19	P20	P21
General Aspects	Studies ID	[SMAP-21]	[SMAP-22]	[SMAP-23], [SMAP-24], [SMAP-53], [SMAP-59], [SMAP-60], [SMAP-61], [SMAP-62], [SMAP-69]
	Year of publication	2004	2004	2010, 2013, 2008, 2011, 2012, 2013, 2013, 2009
	Type of venue	Conference	Journal	Conference, Book Section, Conference, Conference, Workshop, Conference, Conference, Conference
	Author(s) countries	USA	USA	Luxembourg, Mexico, Spain
Overview	Classification of the proposal	Templates	Templates	Patterns with context information
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	Security requirements	Security requirements	General purpose (S)
	Domain generalizable?	No	No	No
	Type of reqs	F, NF	NS	F, NF, NT
	Artefact to reuse	SNL	SNL	SNL
	Notation	Natural language	Natural language	Natural language
Structure	Structure	IK	IK	BS
	Structure information	document templates	parameterized requirements	parameterized requirements sentences + metadata + applicability conditions
Organi-zation	Relationships	No	No	Yes (implies, excludes, damages, contributes)
	Arrangement	RPS	RPS	RPS
	Classification	No	No	Yes
Meta-model	Reuse artefact	Yes	No	Yes
	Arrangement	Yes	No	Yes
Me-thods	Construction	Partial	No	Yes
	Reuse	Partial	Yes	Yes
	Update	Partial	No	Yes
Cata-logue	Exists?	Yes	No	Yes
	Domain	Project	---	General
	State	NS	---	Finished
Tools & Validation	Related tools	Introducing and consulting data from the DB.	No	Managing patterns (CRUD), managing classifications schemas (CRUD), creating SRSs using patterns and managing them (CRUD).
	Tests	Yes	No	No
	Type of research	Non-empirical	Non-empirical	Empirical

	Proposal ID	P22	P23	P24
General Aspects	Studies ID	[SMAP-25]	[SMAP-26], [SMAP-66]	[SMAP-27]
	Year of publication	2005	1998, 1997	2014
	Type of venue	Conference	Conference, Conference	Conference
	Author(s) countries	Brazil	Germany	Korea
Overview	Classification of the proposal	Patterns with context information	Formalization/ Specification patterns	Domain/NF patterns
	Scope	Elicitation, Specification	Specification	Specification
	Domain	Geographic information systems	General purpose (S)	Adaptive human management systems
	Domain generalizable?	No	No	No
	Type of reqs	F	NS	F, NF, NT
	Artefact to reuse	CM	Formalization of requirements	UC, Architecture design
	Notation	UML-Geo Frame	Temporal formal logic	Natural language, UML
Structure	Structure	BS	BS	BS
	Structure information	requirements (model) + metadata + applicability conditions	requirement + formalization + metadata + applicability conditions	use case diagram + use case description (parameterized sentences) + architecture design + other stages info
Organi -zation	Relationships	Yes (generalization, specialization, aggregation, composition or association)	No	No
	Arrangement	TX	RPS	NS
	Classification	Yes	No	NS
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	No	Yes
	Reuse	Partial	No	No
	Update	NO (but update is considered)	Partial	No
Cata-logue	Exists?	No	No	Yes
	Domain	---	---	General
	State	---	---	NS
Tools & Validation	Related tools	Using the patterns catalogue (searching on it), using a pattern in a project, creating new patterns, and creating UML-GeoFrame conceptual models.	No	No
	Tests	No	No	No
	Type of research	Non-empirical	Non-empirical	Non-empirical

	Proposal ID	P25	P26	P27
General Aspects	Studies ID	[SMAP-28], [SMAP-29]	[SMAP-30]	[SMAP-31], [SMAP-32], [SMAP-94]
	Year of publication	2005, 2006	2012	2006, 2006, 2004
	Type of venue	Journal, Journal	Conference	Master Thesis paper, Master Thesis paper, Conference
	Author(s) countries	Germany	Denmark	Belgium, USA
Overview	Classification of the proposal	Process patterns	Templates	Domain/NF patterns
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation
	Domain	General purpose (NS)	Embedded systems	Security requirements
	Domain generalizable?	No	No	No
	Type of reqs	---	NS	F, NF, NT
	Artefact to reuse	RE practices	SNL	Formalization of require-ments, Threat tree models
	Notation	Natural language	Natural language	Temporal formal logic, Threat trees
Structure	Structure	BS	IK	BS
	Structure information	requirement + metadata + applicability conditions	parameterized requirement sentence	formalization + threat tree model + metadata
Organi-zation	Relationships	Yes (field Related Patterns, no types specified)	No	No
	Arrangement	RPS	RPS	NS
	Classification	No	No	NS
Meta-model	Reuse artefact	Yes	No	No
	Arrangement	No	No	No
Me-thods	Construction	Yes	No	Yes
	Reuse	No	No	Yes
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	General	Project	General
	State	Evolution	NS	Evolution
Tools & Validation	Related tools	Web tool for searching patterns in the repository and browsing them.	No	No
	Tests	Yes	Yes	Yes
	Type of research	Non-empirical	Experience report	Non-empirical

	Proposal ID	P28	P29	P30
General Aspects	Studies ID	[SMAP-33]	[SMAP-34]	[SMAP-35]
	Year of publication	2003	2012	2014
	Type of venue	Conference	Workshop	Book Section
	Author(s) countries	Germany	Germany	Germany
Overview	Classification of the proposal	Patterns with context information	Patterns with context information	Patterns with context information
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	Security requirements	Law requirements	General purpose (NS)
	Domain generalizable?	No	No	No
	Type of reqs	F	F, NF, NT	F, NF, NT
	Artefact to reuse	SNL	SNL	SNL
	Notation	Natural language	Natural language	Natural language
Structure	Structure	BS	BS	BS
	Structure information	parameterized requirements + metadata	requirement sentence + metadata	requirement sentence + metadata + applicability conditions
Organi-zation	Relationships	No	No	Yes (field Relations, no types specified)
	Arrangement	RPS	TX	NS
	Classification	No	No	NS
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	Yes	No
	Reuse	No	Partial	Yes
	Update	No	Partial	Yes
Cata-logue	Exists?	Yes	Yes	No
	Domain	Project	Project	---
	State	NS	NS	---
Tools & Validation	Related tools	Structuring the requirements in the SRSs, helping in that way the navigation between patterns.	No	No
	Tests	Yes	No	No
	Type of research	Experience report	Non-empirical	Non-empirical

	Proposal ID	P31	P32	P33
General Aspects	Studies ID	[SMAP-36]	[SMAP-37]	[SMAP-38]
	Year of publication	2010	2011	2009
	Type of venue	Conference	Conference	Conference
	Author(s) countries	Jordan	Colombia	Norway
Overview	Classification of the proposal	Patterns with context information	Templates	Templates, Domain/NF patterns
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	General purpose (S)	General purpose (S)	Security requirements
	Domain generalizable?	No	No	No
	Type of reqs	F, NF	NF	NF
	Artefact to reuse	UC	SNL	SNL
	Notation	NS	Natural language	Natural language
Structure	Structure	BS	IK	IK
	Structure information	use case + associated requirements (F, NF) + metadata + applicability conditions	questions sentences	requirement sentence
Organi-zation	Relationships	Yes (inclusion, exclusion)	No	No
	Arrangement	RPS	NS	TX
	Classification	No	NS	Yes
Meta-model	Reuse artefact	Yes	No	No
	Arrangement	No	No	No
Me-thods	Construction	Yes	No	No
	Reuse	No	Yes	No
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	General	General	Project
	State	Evolution	Finished	Evolution
Tools & Validatio	Related tools	No	No	No
	Tests	Yes	No	No
	Type of research	Empirical	Non-empirical	Non-empirical

	Proposal ID	P34	P35	P36
General Aspects	Studies ID	[SMAP-39], [SMAP-40], [SMAP-41]	[SMAP-42]	[SMAP-43]
	Year of publication	1997, 1997, 1997	2008	2002
	Type of venue	Workshop, Journal, Conference	Conference	Conference
	Author(s) countries	United Kingdom	China	Costa Rica, Spain
Overview	Classification of the proposal	Templates	Writing guidelines patterns	Templates
	Scope	Elicitation, Specification	Specification	Elicitation, Specification
	Domain	General purpose (NS)	Network software	General purpose (NS)
	Domain generalizable?	No	No	No
	Type of reqs	F	F, NF	F, NF, NT
	Artefact to reuse	SNL (grouped according to components)	Structure of SNL	Scenarios, UC, Activity diagrams, Data-flows, Document tasks, Workflows
	Notation	Natural language	Natural language	NS
Structure	Structure	IK	IK	IK
	Structure information	parameterized requirements	requirement	diagram
Organi-zation	Relationships	No	No	Yes (dependency, extension, inclusion, equivalence, subset, exception, complement)
	Arrangement	RPS	NS	NS
	Classification	No	NS	NS
Meta-model	Reuse artefact	No	Yes	Yes
	Arrangement	No	No	No
Me-thods	Construction	Yes	No	No
	Reuse	Partial	No	Yes
	Update	No	No	Partial
Cata-logue	Exists?	Yes	No	No
	Domain	Project	---	---
	State	NS	---	---
Tools & Validation	Related tools	Creating new tools for reusing the patterns (a new tool has to be created for each catalogue or domain).	Eliciting the requirements using the proposed NL templates and formalizing the requirements in NL into models requisites.	Editing and requirements and patterns, analysing the consistency, managing the domain lexicon, and translating diagrams.
	Tests	No	No	No
	Type of research	Empirical	Empirical	Non-empirical

	Proposal ID	P37	P38	P39
General Aspects	Studies ID	[SMAP-44]	[SMAP-45]	[SMAP-46]
	Year of publication	2014	2003	2002
	Type of venue	Conference	Conference	Conference
	Author(s) countries	Turkey	Canada, Brazil	USA
Overview	Classification of the proposal	Templates	Domain/NF patterns	Patterns with context information
	Scope	Elicitation, Specification	Specification	Specification
	Domain	General purpose (NS)	Early requirements (goals)	Embedded systems
	Domain generalizable?	No	No	Yes
	Type of reqs	F, NF, NT	F, NF, NT	F, NF
	Artefact to reuse	SNL, Ontologies	Organizational structure	CM, UC, Sequence diagrams, SNL
	Notation	Natural language, OWL	*i**, TELOS	Natural language, UML (something similar to UML)
Structure	Structure	IK	BS	BS
	Structure information	parameterized requirements + ontology	*i** models + formalization	class diagrams + metadata + applicability conditions + other stages information
Organi-zation	Relationships	No	No	Yes (field Related Patterns, no types specified)
	Arrangement	RPS	NS	RPS
	Classification	Yes	NS	No
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	No	No
	Reuse	No	No	No
	Update	Partial	No	No
Cata-logue	Exists?	Yes	No	Yes
	Domain	Project	---	General
	State	NS	---	NS
Tools & Validation	Related tools	Tool based on an interactive product configuration scenario which guides the user to enter required information to instantiate a valid product of the product line and the related requirements with this configuration is enlisted automatically.	No	No
	Tests	Yes	No	No
	Type of research	Empirical	Non-empirical	Non-empirical

	Proposal ID	P40	P41	P42
General Aspects	Studies ID	[SMAP-47]	[SMAP-48]	[SMAP-49]
	Year of publication	2005	2012	2006
	Type of venue	Conference	Conference	Conference
	Author(s) countries	USA	Spain, USA	United Kingdom
Overview	Classification of the proposal	Patterns with context information, Formalization/Specification patterns	Patterns with context information	Domain/NF patterns
	Scope	Elicitation, Specification	Elicitation, Specification	Specification
	Domain	Embedded systems	General purpose (NS)	Service-oriented information exchange requirements
	Domain generalizable?	No	No	No
	Type of reqs	F	F, NF	F, NF, NT
	Artefact to reuse	SNL	SNL, Business process models	SNL, CM
	Notation	Natural language, temporal formal logic	Natural language, A own one (similar to business process modelling languages)	Natural language, UML
Structure	Structure	BS	BS	BS
	Structure information	requirement stated using a structured English grammar + formalization + metadata + applicability conditions	requirements + metadata + diagram + other stages information	class diagrams + questions + metadata + applicability conditions
Organi-zation	Relationships	Yes (field Related Patterns, no types specified)	Yes (shown in a diagram inside the pattern, no types specified)	Yes (achievedBy, changes, expires, followedBy, guards, has, hasAlternate, implies, mayCompose, mayInvolve, mayRequire, maySpecify, references, requiredFor)
	Arrangement	TX	NS	TX
	Classification	Yes	NS	No
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	No	No
	Reuse	No	No	No
	Update	No	No	No
Cata-logue	Exists?	Yes	No	Yes
	Domain	General	---	General
	State	Evolution	---	Evolution
Tools & Validatio	Related tools	No	No	No
	Tests	No	Yes	No
	Type of research	Non-empirical	Non-empirical	Non-empirical

General Aspects	Proposal ID	P43	P44	P45
	Studies ID	[SMAP-50], [SMAP-51]	[SMAP-52]	[SMAP-54]
	Year of publication	1998, 1999	2008	2008
	Type of venue	Journal, Conference	Journal	Conference
	Author(s) countries	United Kingdom, Austria, Germany	Austria, United Kingdom	Spain
Overview	Classification of the proposal	Patterns with context information	Patterns with context information	Templates
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	General purpose (S)	General purpose (NS)	General purpose (NS)
	Domain generalizable?	No	No	No
	Type of reqs	F	F, NF, NT	F, NF, NT
	Artefact to reuse	SNL (grouped into family models, i.e. viewpoints)	SNL	SNL
	Notation	Natural language	Natural language	Natural language
Structure	Structure	BS	BS	IK
	Structure information	parameterized reqs sentences + metadata + applicability conditions	requirements + metadata	requirement
Organi-zation	Relationships	Yes (could be any string e.g. applies_to, depends_on, opposite_to)	Yes (simple, extended, refinedBy, satisifedBy)	No
	Arrangement	RPS	NS	NS
	Classification	No	NS	NS
Meta-model	Reuse artefact	Yes	Yes	No
	Arrangement	No	No	No
Me-thods	Construction	Partial	Yes	Partial
	Reuse	Yes	Yes	Partial
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	Project	Project	Project
	State	NS	NS	NS
Tools & Validation	Related tools	Database to manage viewpoint and requirements.	No	Reusing reqs from previous SRSs: the tool allows selecting for each new SRS what are the reqs that can be reused in other projects, as well as creating new SRSs creating 3 types of reqs.
	Tests	No	Yes	Yes
	Type of research	Empirical	Empirical	Empirical

	Proposal ID	P46	P47	P48
General Aspects	Studies ID	[SMAP-55]	[SMAP-56]	[SMAP-57]
	Year of publication	2013	2014	2011
	Type of venue	Journal	Conference	Conference
	Author(s) countries	Spain	Norway	Japan
Overview	Classification of the proposal	Templates	Templates	Patterns with context information
	Scope	Elicitation, Specification	Elicitation, Specification	Specification
	Domain	Home automation systems	Maintenance and maintainability of rail networks in Europe	Security requirements
	Domain generalizable?	No	No	No
	Type of reqs	NS	NF	F, NF, NT
	Artefact to reuse	SNL	SNL	Misuse cases
	Notation	Natural language	Natural language	Natural language, UML
Structure	Structure	IK	IK	BS
	Structure information	parameterized requirement sentence	requirement sentence	misuse case + metadata + applicability conditions + other stages information
Organi-zation	Relationships	Yes (traceability relationships, no types specified)	Yes (no types specified)	No
	Arrangement	NS	NS	NS
	Classification	NS	NS	NS
Meta-model	Reuse artefact	Yes	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	Yes	No
	Reuse	Yes	No	Yes
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	Project	General	Project
	State	NS	NS	NS
Tools & Validation	Related tools	REMM+, to manage and use templates, define traceability links, direct transformation from requirements specifications produced with templates to a requirements model.	IBM Rational Doors and Excel have been modified to implement the CoVeR method, which ensures that all the regulatory requirements are extracted, adapted and that the result from the work can be reused.	No
	Tests	Yes	Yes	Yes
	Type of research	Non-empirical	Empirical	Empirical

	Proposal ID	P49	P50	P51
General Aspects	Studies ID	[SMAP-58]	[SMAP-63]	[SMAP-64], [SMAP-65]
	Year of publication	2015	2015	2014, 2015
	Type of venue	Journal	Conference	Journal, Journal
	Author(s) countries	Mexico, Spain	USA	Iran
Overview	Classification of the proposal	Patterns with context information	Templates	Formalization/Specification patterns
	Scope	Elicitation, Specification	Elicitation, Specification	Specification
	Domain	General purpose (NS)	General purpose (NS)	General purpose (NS)
	Domain generalizable?	No	No	No
	Type of reqs	F, NF, NT	F, NF, NT	F
	Artefact to reuse	SNL	SNL	Activity diagrams
	Notation	Natural language	Natural language	UML
Structure	Structure	BS	IK	BS
	Structure information	parameterized requirements sentences+ metadata + applicability conditions	requirement	ontologies + use case diagram + activity diagrams
Organi-zation	Relationships	No	No	No
	Arrangement	TX	TX	RPS
	Classification	Yes	Yes	No
Meta-model	Reuse artefact	No	No	Yes
	Arrangement	No	No	No
Me-thods	Construction	Yes	Yes	Yes
	Reuse	Yes	Yes	No
	Update	Yes	No	No
Cata-logue	Exists?	No	Yes	Yes
	Domain	---	Project	Project
	State	---	NS	NS
Tools & Validation	Related tools	Requisite Pro plugin for: managing user roles, creating the catalogue, searching in the catalogue, select reusable reqs, validate reusable reqs, adapt reusable reqs, and managing the catalogue.	No	Prototype tool for extracting models and repository knowledge, and recommending activity diagrams to reuse and adapting these models to a specific project.
	Tests	Yes	Yes	Yes
	Type of research	Empirical	Experience report	Empirical

	Proposal ID	P52	P53	P54
General Aspects	Studies ID	[SMAP-67]	[SMAP-68]	[SMAP-70], [SMAP-71]
	Year of publication	2011	2011	2014, 2014
	Type of venue	Conference	Workshop	Conference, Conference
	Author(s) countries	Austria	Germany	USA
Overview	Classification of the proposal	Formalization/Specification patterns	Writing guidelines patterns	Patterns with context information
	Scope	Specification	Specification	Elicitation, Specification
	Domain	General purpose (NS)	General purpose (NS)	Security requirements
	Domain generalizable?	No	No	No
	Type of reqs	NS	NS	F, NF, NT
	Artefact to reuse	Evolution patterns	Structure of SNL	SNL
	Notation	A own one	Natural language	Natural language
Structure	Structure	BS	IK	BS
	Structure information	diagram + metadata + applicability conditions	SNL structure	parameterized requirements sentences + applicability conditions
Organi-zation	Relationships	Yes (no types specified)	No	No
	Arrangement	NS	NS	RPS
	Classification	NS	NS	Yes
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	No	Partial
	Reuse	Partial	No	Yes
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	General	General	General
	State	NS	Finished	NS
Tools & Validation	Related tools	No	No	Taking as input reqs-related artefacts (SRSs, feature requests, use case scenarios etc.), it generates a set of security requirements, automatically presenting a list of applicable templates that can be selected and instantiated.
	Tests	No	Yes	Yes
	Type of research	Non-empirical	Empirical	Empirical

	Proposal ID	P55	P56	P57
General Aspects	Studies ID	[SMAP-72], [SMAP-73]	[SMAP-74]	[SMAP-75]
	Year of publication	2011, 2012	2013	1999
	Type of venue	Workshop, Workshop	Conference	Conference
	Author(s) countries	Germany	USA	Japan
Overview	Classification of the proposal	Patterns with context information	Patterns with context information	Patterns with context information
	Scope	Elicitation, Specification	Elicitation	Elicitation, Specification
	Domain	Usability requirements	Sustainability requirements	General purpose (S)
	Domain generalizable?	No	No	No
	Type of reqs	F	F, NF, NT	F
	Artefact to reuse	SNL	SNL	Interviews, Questionnaires, UC
	Notation	Natural language	Natural language	UML
Structure	Structure	BS	BS	BS
	Structure information	guidelines + metadata + applicability conditions	requirements + metadata + applicability conditions	questions sentences + use case diagrams
Organi-zation	Relationships	No	Yes (field Related Patterns, no types specified)	Yes (only between use case patterns)
	Arrangement	NS	NS	RPS
	Classification	NS	NS	No
Meta-model	Reuse artefact	No	No	Yes
	Arrangement	No	No	No
Me-thods	Construction	No	Partial	No
	Reuse	No	No	Partial
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	No
	Domain	General	General	---
	State	NS	Evolution	---
Tools & Validation	Related tools	TULIP tool: extended use case editor developed as a supporting tool, allowing adding annotations to use case elements (as defined by annotation templates).	No	No
	Tests	No	No	No
	Type of research	Non-empirical	Non-empirical	Non-empirical

	Proposal ID	P58	P59	P60
General Aspects	Studies ID	[SMAP-76]	[SMAP-77]	[SMAP-78]
	Year of publication	2012	2015	2014
	Type of venue	Journal	Conference	Conference
	Author(s) countries	India	Brazil	Austria
Overview	Classification of the proposal	Patterns with context information	Process patterns	Patterns with context information
	Scope	Elicitation, Specification	Specification	Elicitation, Specification
	Domain	Security requirements	Reqs for SRSs (related to correctness, consistency and completeness)	General purpose (NS)
	Domain generalizable?	No	No	No
	Type of reqs	F, NF, NT	---	F, NF, NT
	Artefact to reuse	SNL, Ontologies	Refinement models (in *i** to achieve NFRs)	SNL
	Notation	Natural language, Ontology language (NS)	*i**	Natural language
Structure	Structure	BS	BS	BS
	Structure information	parameterized reqs + metadata + ontology + applicability conditions	*i** model + metadata + applicability conditions	requirement sentence + metadata
Organi-zation	Relationships	Yes (field Related Patterns, no types specified)	Yes (specialization-of, part-of, occurrence-of)	No
	Arrangement	NS	NS	RPS
	Classification	NS	NS	No
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	Partial	No	No
	Reuse	Partial	Partial	Partial
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	No
	Domain	Project	General	---
	State	NS	NS	---
Tools & Validation	Related tools	No	No	Instantiating patterns; creating SRSs; searching/ browsing patterns; adding new reqs (not coming from patterns); while the user is entering a new req, proposing similar patterns or reqs in other SRSs.
	Tests	No	No	No
	Type of research	Non-empirical	Non-empirical	Non-empirical

General Aspects	Proposal ID	P61	P62	P63
	Studies ID	[SMAP-79]	[SMAP-80]	[SMAP-81]
	Year of publication	2007	2010	2012
	Type of venue	Conference	Conference	Conference
	Author(s) countries	Korea	Sweden	Portugal
Overview	Classification of the proposal	Patterns with context information	Patterns with context information	Patterns with context information
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	Embedded systems	General purpose (S)	Geospatial requirements
	Domain generalizable?	Yes	No	No
	Type of reqs	F, NF, NT	F, NF	F, NF
	Artefact to reuse	SNL	CM, UC, Sequence diagrams, SNL	SNL, Feature models, CM, Sequence diagrams
	Notation	Natural language	Natural language, UML (something similar to UML)	Natural language, MATA
Structure	Structure	BS	BS	BS
	Structure information	F requirements sentences + quality requirements + metadata + applicability conditions + other stages information	class diagrams + metadata + applicability conditions + other stages info	requirements sentences + metadata + feature model + class diagram + sequence diagram + applicability conditions
Organi-zation	Relationships	Yes (field Related Features, no types specified)	Yes (field Related Patterns, no types specified)	Yes (field Related Patterns, no types specified)
	Arrangement	RPS	NS	NS
	Classification	Yes	NS	NS
Meta-model	Reuse artefact	Yes	No	No
	Arrangement	Yes	No	No
Me-thods	Construction	Yes	No	No
	Reuse	Yes	No	No
	Update	Partial	No	No
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	Project	General	General
	State	NS	NS	NS
Tools & Validation	Related tools	No	No	Managing patterns using the Eclipse environment.
	Tests	Yes	No	No
	Type of research	Empirical	Non-empirical	Non-empirical

	Proposal ID	P64	P65	P66
General Aspects	Studies ID	[SMAP-82]	[SMAP-83]	[SMAP-84]
	Year of publication	2003	2014	2013
	Type of venue	Workshop	Conference	Journal
	Author(s) countries	Norway	USA	India
Overview	Classification of the proposal	Templates	Patterns with context information	Patterns with context information
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	Security requirements	Security requirements	Online examination systems
	Domain generalizable?	No	No	Yes
	Type of reqs	F, NF, NT	F, NF, NT	F, NF, NT
	Artefact to reuse	Misuse cases, UC, SNL	SNL, Feature models	SNL
	Notation	Natural language	Natural language, Modelling Language (NS)	Natural language
Structure	Structure	IK	BS	BS
	Structure information	parameterized requirements sentences for: misuse cases, use cases and requirements	questions + metadata + applicability conditions	requirement sentence + metadata + applicability conditions
Organi-zation	Relationships	No	Yes (expressed in a feature diagram)	Yes (fields Extra Requirements and Related Patterns, no types specified)
	Arrangement	NS	NS	RPS
	Classification	NS	NS	Yes
Meta-model	Reuse artefact	Yes	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	Yes	Partial
	Reuse	Yes	No	Yes
	Update	No	No	No
Cata-logue	Exists?	No	Yes	Yes
	Domain	---	General	Project
	State	---	NS	Finished
Tools & Validatio	Related tools	No	No	No
	Tests	No	Yes	No
	Type of research	Non-empirical	Empirical	Non-empirical

	Proposal ID	P67	P68	P69
General Aspects	Studies ID	[SMAP-85]	[SMAP-86], [SMAP-87], [SMAP-88]	[SMAP-89]
	Year of publication	2015	2009, 2010, 2010	2014
	Type of venue	Workshop	Workshop, Conference, Workshop	Journal
	Author(s) countries	Canada	USA, United Kingdom, Brazil	Greece
Overview	Classification of the proposal	Templates	Domain/NF patterns	Domain/NF patterns, Templates
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	General purpose (NS)	General purpose (S)	eGov public services provision systems
	Domain generalizable?	No	No	No
	Type of reqs	F, NF, NT	NF	F, NF, NT
	Artefact to reuse	SNL	Refinement models (in *i** to achieve NFRs)	SNL, UC
	Notation	Natural language	*i**	Natural language, UML
Structure	Structure	IK	BS	IK
	Structure information	questions	*i** model + metadata + applicability conditions	requirements sentences + use cases
Organi-zation	Relationships	No	Yes (specialization-of, part-of, occurrence-of)	Yes (aggregation, influence; not all types specified)
	Arrangement	TX	NS	RPS
	Classification	Yes	NS	Yes
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	No	Yes
	Reuse	No	Yes	No
	Update	No	No	No
Cata-logue	Exists?	Yes	No	Yes
	Domain	General	---	General
	State	NS	---	NS
Tools & Validation	Related tools	No	Tow tools: one for NFRs modelling, another one for capturing and visualizing NFRs.	Use of commercial tools for modelling and storing the reusable knowledge (requirements and use cases).
	Tests	No	Yes	No
	Type of research	Non-empirical	Empirical	Non-empirical

	Proposal ID	P70	P71	P72
General Aspects	Studies ID	[SMAP-90]	[SMAP-91]	[SMAP-92]
	Year of publication	2006	1999	2002
	Type of venue	Conference	Conference	Journal
	Author(s) countries	Malaysia	Spain	Spain
Overview	Classification of the proposal	Writing guidelines patterns	Patterns with context information	Patterns with context information
	Scope	Specification	Elicitation, Specification	Elicitation, Specification
	Domain	General purpose (S)	Information system requirements	Security requirements
	Domain generalizable?	No	Yes	No
	Type of reqs	F, NF, NT	F, NF	F, NF
	Artefact to reuse	Grammars	SNL, UC	SNL
	Notation	Natural language, Grammar language (NS)	Natural language	Natural language
Structure	Structure	BS	BS	BS
	Structure information	SNL structure + guidelines	parameterized requirements + metadata + applicability conditions	parameterized requirements sentences + metadata + applicability conditions + other stages information
Organi-zation	Relationships	No	No	Yes (inclusion, exclusion)
	Arrangement	TX	RPS	RPS
	Classification	Yes	No	Yes
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	No	No	No
	Reuse	No	Partial	Yes
	Update	No	No	Partial
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	General	Project	Project
	State	Finished	NS	Evolution
Tools & Validatio	Related tools	No	No	No
	Tests	Yes	Yes	Yes
	Type of research	Empirical	Non-empirical	Empirical

	Proposal ID	P73	P74	P75
General Aspects	Studies ID	[SMAP-93]	[SMAP-95], [SMAP-96]	[SMAP-98]
	Year of publication	2011	2013, 2013	2002
	Type of venue	Conference	Conference, Conference	Conference
	Author(s) countries	Spain	Ireland	Japan
Overview	Classification of the proposal	Patterns with context information	Templates, Domain/NF patterns	Patterns with context information
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	Learning systems requirements	Autonomy requirements for space missions	Web applications
	Domain generalizable?	No	No	No
	Type of reqs	F, NF	F, NF, NT	F, NF
	Artefact to reuse	SNL	SNL	SNL, UC
	Notation	Natural language	Natural language	Natural language, Modelling Language (NS)
Structure	Structure	BS	IK	BS
	Structure information	parameterized requirements sentences + metadata + applicability conditions + other stages information	requirement sentence	requirements + metadata + applicability conditions + other stages information
Organi-zation	Relationships	Yes (no types specified)	No	Yes (field Related Patterns, no types specified)
	Arrangement	RPS	TX	RPS
	Classification	Yes	Yes	Yes
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	Yes	No	No
	Reuse	Yes	Yes	Yes
	Update	No	No	No
Cata-logue	Exists?	Yes	Yes	Yes
	Domain	NS	General	Project
	State	Evolution	NS	NS
Tools & Validation	Related tools	SIREN tool for patterns management, complementing the RE process provided by RequesitePro (SIREN tool is a plugin of RequesitePro).	No	No
	Tests	No	Yes	No
	Type of research	Non-empirical	Empirical	Non-empirical

	Proposal ID	P76	P77	P78
General Aspects	Studies ID	[SMAP-99], [SMAP-100]	[SMAP-101]	[SMAP-102]
	Year of publication	2001, 2001	2007	2012
	Type of venue	Conference, Workshop	Book	Workshop
	Author(s) countries	Japan	USA	Germany
Overview	Classification of the proposal	Templates, Writing guidelines patterns	Patterns with context information	Patterns with context information
	Scope	Elicitation, Specification	Elicitation, Specification	Elicitation, Specification
	Domain	General purpose (S)	General purpose (S)	Seismology requirements
	Domain generalizable?	No	No	No
	Type of reqs	F	NF, NT	F
	Artefact to reuse	Structure of SNL (for scenarios/use cases)	SNL	SNL, CM
	Notation	Natural language	Natural language	Natural language, UML
Structure	Structure	IK	BS	BS
	Structure information	parameterized requirements sentences + constraints	parameterized requirements + applicability conditions + other stages information	class diagrams + metadata + applicability conditions + other stages information
Organi-zation	Relationships	Yes (among the parts, the case frames, of patterns)	Yes (has, displays, uses, is-a)	No
	Arrangement	RPS	RPS	NS
	Classification	No	Yes	NS
Meta-model	Reuse artefact	No	No	No
	Arrangement	No	No	No
Me-thods	Construction	Partial	Yes	Partial
	Reuse	Yes	Yes	No
	Update	No	Partial	No
Cata-logue	Exists?	Yes	Yes	No
	Domain	Project	General	---
	State	NS	NS	---
Tools & Validatio	Related tools	No	No	No
	Tests	Yes	No	No
	Type of research	Empirical	Non-empirical	Non-empirical

	Proposal ID	P79	P80
General Aspects	Studies ID	[SMAP-103]	[SMAP-104]
	Year of publication	2011	2015
	Type of venue	Workshop	Workshop
	Author(s) countries	China	Canada
Overview	Classification of the proposal	Domain/NF patterns	Domain/NF patterns
	Scope	Elicitation	Elicitation, Specification
	Domain	Security requirements	Privacy and transparency requirements
	Domain generalizable?	No	No
	Type of reqs	NS	NF
	Artefact to reuse	Problem frames	Refinement models (in *i** to achieve NFRs)
	Notation	*i**	*i**
Structure	Structure	BS	BS
	Structure information	*i** model + metadata	*i** model + metadata
Organi-zation	Relationships	No	Yes (specialization-of, part-of, occurrence-of)
	Arrangement	NS	NS
	Classification	NS	NS
Meta-model	Reuse artefact	No	No
	Arrangement	No	No
Me-thods	Construction	Partial	Yes
	Reuse	Partial	Partial
	Update	No	No
Cata-logue	Exists?	Yes	Yes
	Domain	General	General
	State	NS	Evolution
Tools & Validatio	Related tools	No	No
	Tests	No	No
	Type of research	Non-empirical	Non-empirical

Appendix III – SRP Metamodel

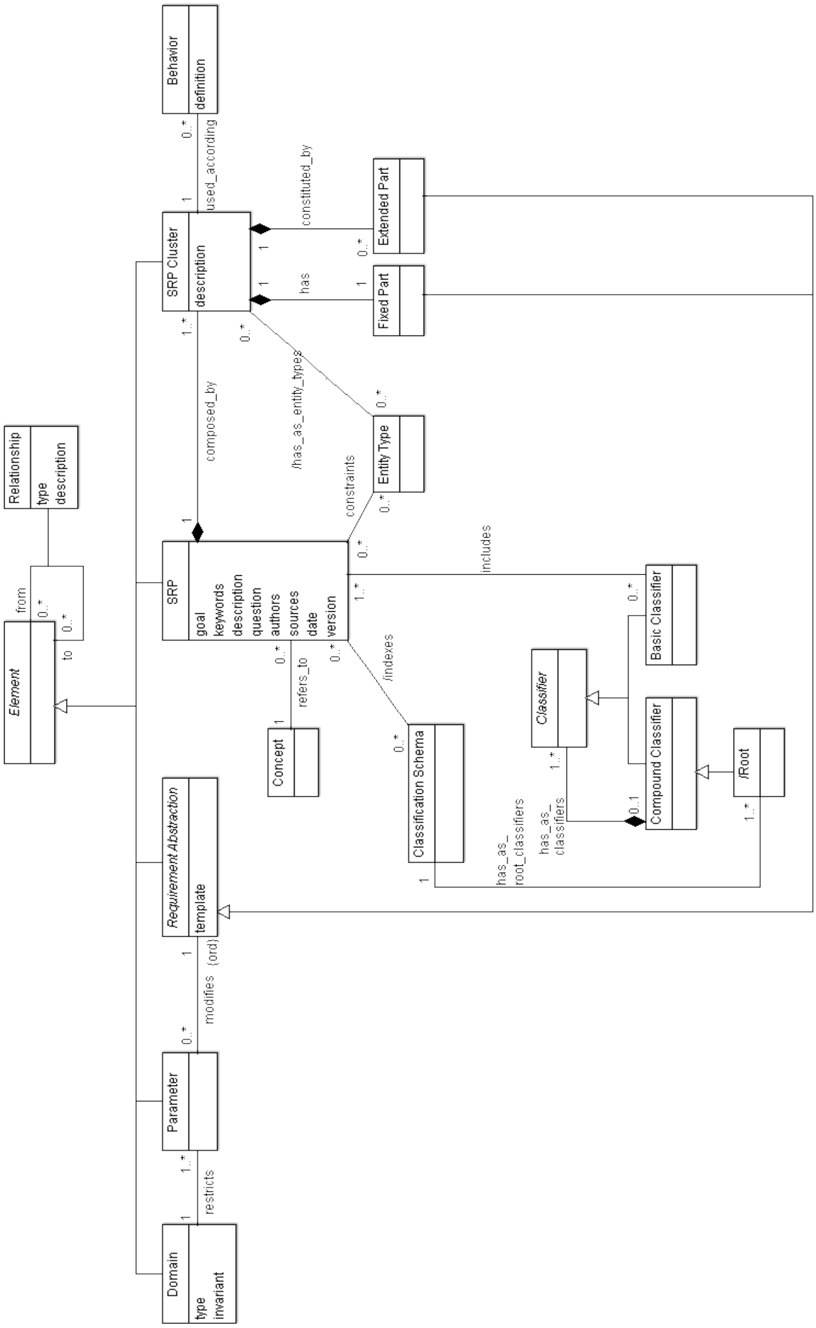
This appendix contains the whole metamodel of SRPs and their organization in a catalogue, which is presented in three parts in Chapter 4, as well as the glossary with all the concepts appearing on it.

Glossary of the SRP Metamodel

- **Basic classifier.** Category in the lower level of a classification schema (i.e. indexing SRPs).
- **Behavior.** Constraint to which the fixed and extended parts of an SRP cluster must conform to.
- **Classification schema.** Taxonomy that organizes SRPs.
- **Classifier.** Category in a classification schema.
- **Compound classifier.** Category in the middle level of a classification schema (i.e. containing other classifiers); it is used to create a hierarchical structure of classifiers.
- **Concept.** Main aspect to which an SRP refers to.
- **Domain.** Valid values that a parameter can take when applying in a project the requirement abstraction to which the parameter is associated.
- **Element.** Generalization of the atomic components of SRPs that could be involved in relationships.
- **Entity type.** General aspect of an IT project restricted by an SRP.
- **Extended part.** Requirement abstraction of an SRP cluster that allows defining a precise requirement, providing more detail to the fixed part of that same SRP cluster.
- **Fixed part.** Requirement abstraction of an SRP cluster that allows defining the minimal requirement that always holds in the SRP cluster.
- **Parameter.** Variable part in a requirement abstraction that takes a specific value of its domain during the application of that requirement abstraction in a project.
- **Relationship.** Connection, association, or involvement among two different elements of SRPs.
- **Requirement abstraction.** Generalization of the parts of an SRP cluster, which may be a fixed or extended part.
- **Root.** Category in the upper level of a classification schema.
- **SRP.** Pattern that, when applied in a project, produces software requirements that foster the achievement of the goal of that pattern.

- **SRP cluster.** Group of requirement abstractions that allow defining requirements to achieve the goal of that SRP using a specific solution.

SRP Metamodel



Appendix IV – Interview Guide

This appendix shows the interview guide used to perform the interviews of the empirical study presented in Chapter 11.

INTERVIEW GUIDE: STUDY OF REQUIREMENTS REUSE KNOWLEDGE IN SOFTWARE DEVELOPMENT PROJECTS

A. About the interviewee

***Answered before the interview (if possible) ***

With the following questions we want to know personal and experience aspects of the interviewee in order to better understand your answers.

We will not use this information to any other finality than the one described before, and they will not be published as part of the results of this study (it is not mandatory to fill all fields).

A.1 Personal information

Name and surname:

Contact e-mail:

A.2 Studies

Educational background:

Subject:

Professional certificates:

A.3 Professional experience

Years in industry:

Years in university or research labs:

A.4 Professional experience in the organization

Position:

Years in this position:

Years in the organization:

A.5 About the organization

Name of the organization:	
Number of employees:	
Principal production:	
Certifications:	

A.6 About the project

Name of the project:	
Domain of the project:	
Number of participants:	
Project duration:	
Finalization date:	
Economic costs:	

B. Clarifications on "About the interviewee"

**** Answered during the interview ****

The following questions are clarifications on some of the aspects asked at the section "About the Interviewee".

Q0a. If the organization has some certification, what is the maturity level of its process development? *(select one):*

Very low ☐ Low ☐ Medium ☐ High ☐ Very high ☐

Q0b. Clarifications on the selected project (its domain, lifecycle, and so on).

C. About the project

**** Answered during the interview ****

The following questions are about the concrete aspects of our investigation. In this first part of the interview, we want to know about the practices and needs related to requirements. In particular, we want to obtain the conception of the main aspects related to requirements engineering, a profile of the requirements methodology used during the selected project and the challenges faced in relation to requirements, and if any kind of requirements reused was carried out at the project.

C.1 General description of the organization

> *The answer to the following questions might represent the whole requirements engineering team of the selected project.*

Q1. What were the five most important activities in Requirements Engineering (RE) as it was relevant for you and your organization?

Q2. Does the role of "requirements engineer" exist in your company? Is this role played by some designated person or instead it is a hat that a person wears at some moment, and later this very person may become e.g. tester? What are the main responsibilities of this role?

Q3. What do you understand by requirements in your organization?

C.2 General RE practices applied on the project

The answer to the following questions might focus only on the selected project.

Q4. How were requirements elicited?

Q5. What languages were used for specifying requirements?

Q6. Did you follow any standard or template to write requirements specifications? Which ones?

Q7. Did you organize your requirements specification somehow? How? What headers did you have? Were these headers based on something? Are these headers always the same?

Q8. Did you use any tools for the activities in RE? Which ones? For what is each tool used?

Q9. Were there any challenges faced in the RE stage pertaining to elicitation? Which ones? What were the impacts and relevance of these challenges? (relevance from 1..5)

Q10. Were there any challenges faced in the RE stage pertaining to specification? Which ones? What were the impacts and relevance of these challenges? (relevance from 1..5)

C.3 Requirements reuse practices applied on the project

The answer to the following questions might focus only on the selected project.

Q11. In that process of eliciting and specifying requirements, did you use previous requirements from other projects as input for this projects elicitation? Did you also reuse how the requirements were formulated based on previous requirements?

Q12. If answer was "yes" (to first question at Q11), what was the percentage of reused requirements? How do you consider this percentage (low, medium, high, very high)?

Q13. If answer was "yes" (to second question at Q11), what was the percentage of effort reused?

Q14. If answer was "yes" (to any question at Q11), do you think that there was any factor that influenced the level of reuse of requirements? Which ones (some factors from the organization, from the project, from the RE process used, from the requirements engineers themselves, etc.)? Why (not)?

Q15. More concretely (to Q14), did you think that the type of requirement influenced if requirements were reused?

How could this be applied on the project we were talking about?

Was there any particular type of requirements that were more prone to be reused from previous projects because of their type? Which ones? Why?

From the requirements resulting from the project, do you think that there is any particular type of requirement more prone to be reused in future projects? Which ones? Why?

Q16. If answer was "yes" (to any question at Q11), how were requirements reused? Did you use any particular technique to achieve such reuse?

Q17. If answer was "yes" (to any question at Q11), did you use any tool for supporting such reuse? Which ones? Were they specific tools for reusing requirements, or were they the same tools as the ones used for requirements (see Q8)?

Q18. If answer was "yes" (to any question at Q11), how was the requirements reuse process chosen?

Q19. If answer was "yes" (to any question at Q11), did you think that there was any factor influencing the choice of the requirements reuse process? Which ones (some factors from the organization, from the project, from the RE process used, from the requirements engineers themselves, etc.)? Why (not)?

Q20. Do you think requirements reuse was (or would have been) beneficial on the project at hand?

Would more reuse have been beneficial?

If yes, why?

If yes, how could you have invested in reuse in terms of:

- preparatory work?

- during the project?

If yes, why was more reuse not applied?

D. About your opinion on the application of requirements reuse practices in organizations

**** Answered during the interview ****

In the second part of the interview you are asked about your opinion on the current application of requirements reuse practices in organizations according to your knowledge and experience.

Q21. Do you think requirements reuse is an established practice in organizations? Why (not)?

Q22. Do you know about any (other) method to reuse requirements?

If yes, what are the pros and cons of them?

If yes, why did you not use them?

If yes, is something missing on them to incorporate them in your organization? What exactly is missing?

E. About software requirement patterns

**** Answered during the interview ****

In the last part of the interview, we are interested on your opinion on how the maintenance of catalogues or libraries of Software Requirements Patterns (SRPs) could contribute to requirements reuse as well as the critical success factors of their application.

In a nutshell, an SRP consists on natural language templates for generating those requirements that are related to a specific objective (goal), as well as some information to identify its adequacy to a particular project and how it may be tailored to the project. An SRP catalogue, together with the adequate reuse processes and tool support, may exist to facilitate the classification, search and recommendation of suitable SRPs for a specific project.

> *For the answers of these questions, you might imagine that the requirements you thought could be reused in the selected project from a(some) previous project(s) are incorporated in an SRP catalogue.*

Q23. Do you think SRPs would enable reuse in your organization? Why (not)?

Q24. If yes (to Q23), how do you imagine they would be integrated in your RE process?

Q25. Would SRPs be relevant for all types of requirements? Why (not)? For which ones would they be relevant?

Q26. Would there be any pros of using SRPs? If yes, which ones? Why (not)?

Q27. Would there be any cons of using SRPs? If yes, which ones? Why (not)?

Q28. Would there be any factors that could help in the introduction of an SRP catalogue in your organization? If yes, which ones? Why (not)?

Q29. Would there be any factors that could represent a barrier for the introduction of an SRP catalogue in your organization? If yes, which ones? Why (not)?

Appendix V – Interviews Code Relationships

This appendix contains a summary of the categories of the answers from each respondent of the interview-based empirical study presented in Chapter 11. In the chapter before mentioned, the discussion and the findings are based on the data provided in this appendix. By providing the following tables, the reader will be able to verify the discussions and findings from the chapter and assess other potential relationships that are not related to the research question addressed. The first column shows the respondent code and the subsequent columns show the category codes (introduced and detailed in Section 11.4) that each respondent answered.

Respondent	SRPs enable reuse (definition of the categories in Table 11.5 & Table 11.6)	SRP integration (definition of the categories in Table 11.7)	SRPs valid for all req. types (definition of the categories in Table 11.8 & Table 11.9)	SRP pros (definition of the categories in Table 11.10)	SRP cons (definition of the categories in Table 11.11)	SRP introduction factors (definition of the categories in Table 11.12)	SRP introduction barriers (definition of the categories inTable 11.13)
A	Use-No, Use-D, Use-F	–	ReqTy-Yes, ReqTy-A, ReqTy-B, ReqTy-C	Pro-A, Pro-D, Pro-E, Pro-I, Pro-K, Pro-O	Con-I, Con-M	Intr-A, Intr-G	Bar-C, Bar-D, Bar-G, Bar-J
B	Use-Yes, Use-A, Use-H	Inte-A, Inte-B, Inte-C, Inte-D, Inte-E, Inte-G	ReqTy-No, ReqTy-A, ReqTy-B	Pro-A, Pro-C, Pro-H	Con-C, Con-D, Con-E, Con-F, Con-G	Intr-A, Intr-B, Intr-D, Intr-E, Intr-F	Bar-A
C	Use-Yes, Use-B	Inte-A, Inte-B, Inte-E, Inte-G	ReqTy-Yes, ReqTy-A, ReqTy-B, ReqTy-C	Pro-A, Pro-B, Pro-I, Pro-J, Pro-K	Con-E, Con-G	Intr-A, Intr-B, Intr-E	Bar-D, Bar-E
D	Use-Yes, Use-A, Use-J	Inte-B	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-J	Pro-A, Pro-E	Con-A	Intr-A, Intr-B, Intr-D, Intr-E	Bar-C, Bar-D
E	Use-Yes, Use-B, Use-C	Inte-A, Inte-K	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-C, ReqTy-J	Pro-B, Pro-C, Pro-D, Pro-J, Pro-M	Con-B	Intr-C, Intr-D	Bar-C, Bar-D
F	Use-Yes, Use-B, Use-D	Inte-B, Inte-F, Inte-G	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-J	Pro-A, Pro-B, Pro-E, Pro-G, Pro-I	Con-A	Intr-A, Intr-C	Bar-C
G	Use-Yes, Use-C, Use-I	Inte-A, Inte-B, Inte-C	ReqTy-No, ReqTy-A	Pro-E, Pro-N	Con-G, Con-N, Con-O	Intr-A, Intr-C	Bar-C, Bar-E, Bar-F, Bar-H
H	Use-Yes, Use-C, Use-D	Inte-A, Inte-J	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-C, ReqTy-J	Pro-A, Pro-B, Pro-C, Pro-E, Pro-J, Pro-P	Con-F, Con-I	Intr-B, Intr-D	Bar-B, Bar-C, Bar-D
I	Use-Yes, Use-B, Use-C, Use-H	Inte-E	ReqTy-No, ReqTy-D, ReqTy-J	Pro-A, Pro-B, Pro-C, Pro-D, Pro-E, Pro-H	Con-A	Intr-A, Intr-B, Intr-C, Intr-D, Intr-E	Bar-G, Bar-H
J	Use-Yes, Use-A	Inte-B, Inte-C	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-J	Pro-B, Pro-G	Con-E, Con-G, Con-L	Intr-A, Intr-B, Intr-C, Intr-D, Intr-E	Bar-C, Bar-D, Bar-E
K	Use-Yes, Use-B, Use-E, Use-H	Inte-A, Inte-B, Inte-C, Inte-E	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-J	Pro-A, Pro-B, Pro-C, Pro-H	Con-D, Con-E, Con-F, Con-H	Intr-C, Intr-D, Intr-E, Intr-G	Bar-C, Bar-I, Bar-K

Respondent	**SRPs enable reuse** (definition of the categories in Table 11.5 & Table 11.6)	**SRP integration** (definition of the categories in Table 11.7)	**SRPs valid for all req. types** (definition of the categories in Table 11.8 & Table 11.9)	**SRP pros** (definition of the categories in Table 11.10)	**SRP cons** (definition of the categories in Table 11.11)	**SRP introduction factors** (definition of the categories in Table 11.12)	**SRP introduction barriers** (definition of the categories inTable 11.13)
L	Use-No, Use-B, Use-C	Inte-A, Inte-B, Inte-C	ReqTy-No, ReqTy-A, ReqTy-D, ReqTy-J	Pro-A, Pro-B, Pro-C	Con-D, Con-H	Intr-A, Intr-C, Intr-G	Bar-C
M	Use-Yes, Use-B, Use-C	Inte-E	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-C, ReqTy-J	Pro-A, Pro-B, Pro-C, Pro-D, Pro-G	Con-D, Con-H, Con-J, Con-K	Intr-D	Bar-D
N	Use-Yes, Use-E	Inte-B, Inte-F	ReqTy-No, ReqTy-D, ReqTy-J	Pro-A, Pro-D	Con-D, Con-F	Intr-A, Intr-B, Intr-C, Intr-F	Bar-C, Bar-E
O	Use-Yes, Use-D	Inte-C, Inte-F	ReqTy-No, ReqTy-D, ReqTy-E, ReqTy-J	Pro-A, Pro-D, Pro-Q	Con-D, Con-F	Intr-A, Intr-B, Intr-F	Bar-C, Bar-D
P	Use-Yes, Use-A, Use-C, Use-D, Use-E, Use-F	Inte-A, Inte-B, Inte-C, Inte-D	ReqTy-No, ReqTy-A, ReqTy-J	Pro-D, Pro-L	Con-A	Intr-A, Intr-B, Intr-C, Intr-E	Bar-E, Bar-F
Q	Use-Yes, Use-A	Inte-A, Inte-C, Inte-D	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-J	Pro-A, Pro-B, Pro-C, Pro-F	Con-D, Con-F, Con-H	Intr-A, Intr-B, Intr-C, Intr-E	Bar-C, Bar-D
R	Use-Yes, Use-B	Inte-A, Inte-B, Inte-D, Inte-E	ReqTy-No, ReqTy-E	Pro-A, Pro-F	Con-E	Intr-A, Intr-D, Intr-F	Bar-D
S	Use-Yes, Use-A, Use-C, Use-F	Inte-A, Inte-D, Inte-E	ReqTy-Yes, ReqTy-A, ReqTy-B, ReqTy-C	Pro-A, Pro-F	Con-E	Intr-B, Intr-C, Intr-D, Intr-F, Intr-I	Bar-D
T	Use-Yes, Use-A, Use-B	Inte-A, Inte-D	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-C, ReqTy-J	Pro-D, Pro-E, Pro-G, Pro-T	Con-A	Intr-A, Intr-B, Intr-D, Intr-H	Bar-C, Bar-D
U	Use-Yes, Use-A, Use-I	Inte-A, Inte-B, Inte-C, Inte-D	ReqTy-No, ReqTy-A, ReqTy-J	Pro-A, Pro-B, Pro-C, Pro-F	Con-G	Intr-A, Intr-B, Intr-C, Intr-F, Intr-J	Bar-D, Bar-E
V	Use-Yes, Use-A, Use-E	Inte-A, Inte-D, Inte-H	ReqTy-No, ReqTy-A, ReqTy-B, ReqTy-J	Pro-A, Pro-B, Pro-C, Pro-F	Con-E, Con-P	Intr-B, Intr-C	Bar-C, Bar-E, Bar-F

SMAP Bibliography

[SMAP-1] A. Alebrahim, M. Heisel, and R. Meis, "A Structured Approach for Eliciting, Modeling, and Using Quality-Related Domain Knowledge", in International Conference on Computational Science and Its Applications (ICCSA), 2014.

[SMAP-2] D. de Almeida Ferreira, and A. Rodrigues da Silva, "RSLingo - An Information Extraction Approach Towards Formal Requirements Specifications", in IEEE Model-Driven Requirements Engineering Workshop (MoDRE), 2012.

[SMAP-3] D. de Almeida Ferreira, and A. Rodrigues da Silva, "RSL-PL: A linguistic pattern language for documenting software requirements", in IEEE International Workshop on Requirements Patterns (RePa), 2013.

[SMAP-4] K. Beckers, and M. Heisel, "A Foundation for Requirements Analysis of Privacy Preserving Software ", in International Cross-Domain Conference and Workshop on Availability, Reliability, and Security (ARES), 2012.

[SMAP-5] F. B. V. Benitti, and R. C. da Silva, "Evaluation of a Systematic Approach to Requirements Reuse", in Journal of Universal Computer Science, 2013.

[SMAP-6] J. Berrocal, J. Garcia-Alonso, C. Vicente-Chicote, and J. M. Murillo, "A Pattern-Based and Model-Driven Approach for Deriving IT System Functional Models from Annotated Business Models", in International conference on Information Systems Development: Improving Enterprise Communication (ISD), 2014.

[SMAP-7] J. Bhatia, R. Sharma, K. Biswas, and S. Ghaisas, "Using Grammatical Knowledge Patterns for structuring requirements specifications", in IEEE International Workshop on Requirements Patterns (RePa), 2013.

[SMAP-8] J. M. Carrillo de Gea, J. Nicolas, J. L. Fernandez Aleman, A. Toval, A. Vizcaino, and C. Ebert, "Reusing Requirements in Global Software Engineering", in Managing Requirements Knowledge Book (MARKBOOK), 2013.

[SMAP-9] A. L. de Castro Leal, J. L. Braga, and S. M. Serra da Cruz, "Cataloguing Provenance-Awareness with Patterns", in IEEE International Workshop on Requirements Patterns (RePa), 2015.

[SMAP-10] L. Chung, and S. Supakkul, "Capturing and Reusing Functional and Non-functional Requirements Knowledge: A Goal-Object Pattern Approach" in IEEE International Conference on Information Reuse and Integration, 2006.

[SMAP-11] H. Cunha, and J. C. Sampaio do Prado Leite, "Reusing non-functional patterns in i* modelling", IEEE International Workshop on Requirements Patterns (RePa), 2014.

[SMAP-12] O. Daramola, G. Sindre, and T. Moser, "Ontology-based support for security requirements", in Workshops On the Move to Meaningful Internet Systems (OTM), 2012.

[SMAP-13] O. Daramola, G. Sindre, and T. Stalhane, "Pattern-based security requirements specification using ontologies and boilerplates" in IEEE International Workshop on Requirements Patterns (RePa), 2012.

[SMAP-14] R. Darimont, and A. van Lamsweerde, "Formal refinement patterns for goal-driven requirements elaboration" in Symposium on Foundations of Software Engineering (SIGSOFT), 1996.

[SMAP-15] J. Dehlinger, and R. Lutz, "A product-line requirements approach to safe reuse in multi-agent systems", in International Workshop on Software Engineering for Large-scale Multi-Agent systems (SELMA), 2005.

[SMAP-16] J. Dehlinger, and R. Lutz, "A Product-Line Approach to Promote Asset Reuse in Multi-agent Systems", in Software Engineering for Multi-Agent Systems Conference, 2006.

[SMAP-17] C. Denger, D. M. Berry, and E. Kamsties, "Higher quality requirements specifications through natural language patterns", in International Conference on Software: Science, Technology and Engineering (SwSTE), 2003.

[SMAP-18] M. Derakhshanmanesh, J. Fox, and J. Ebert, "Requirements-driven incremental adoption of variability management techniques and tools: an industrial experience report", Requirements Engineering Journal, 2013.

[SMAP-19] D. Dietrich, and J. M. Atlee, "A pattern for structuring the behavioural requirements of features of an embedded system", in IEEE International Workshop on Requirements Patterns (RePa), 2013.

[SMAP-20] A. Elgammal, O. Turetken, W. J. van den Heuvel, and M. Papazoglou, "Formalizing and applying compliance patterns for business process compliance", Software and Systems Modeling Journal, 2014.

[SMAP-21] B. Estes, and D. Pritchard, "Managing system design requirements information" in International Carnahan Conference on Security Technology, 2004.

[SMAP-22] D. Firesmith, "Specifying Reusable Security Requirements", Journal of Object Technology, 2004.

[SMAP-23] X. Franch, C. Palomares, C. Quer, S. Renault, and F. De Lazzer, "A Metamodel for Software Requirement Patterns" in International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2010.

[SMAP-24] X. Franch, C. Quer, S. Renault, C. Guerlain, and C. Palomares, "Constructing and Using Software Requirement Patterns", in Managing Requirements Knowledge Book (MARKBOOK), 2013.

[SMAP-25] V. de Freitas Sodre, L.F. Jugurta, V. M. Vilela, M. Andrade, and V. Marcus, "Improving productivity and quality of GIS databases design using an analysis pattern catalog", in Asia-Pacific conference on Conceptual modelling (APCCM), 2005.

[SMAP-26] R. Gotzhein, M. Kronenburg, and C. Peper, "Reuse in requirements engineering: Discovery and application of a real-time requirement pattern", in Formal Techniques in Real-Time and Fault-Tolerant Systems Conference, 1998.

[SMAP-27] K. Haeng-Kon, "Effective Domain Modeling for Mobile Business AHMS (Adaptive Human Management Systems) requirements", in International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014.

[SMAP-28] L. Hagge, and K. Lappe, "Sharing requirements engineering experience using patterns", IEEE Software, 2005.

[SMAP-29] L. Hagge, and K. Lappe, "Using Requirements Engineering (RE) Patterns for Organizational Learning", Journal of Universal Knowledge Management, 2006.

[SMAP-30] D. Hauksdottir, A. Vermehren, and J. Savolainen, "Requirements reuse at Danfoss" in IEEE International Requirements Engineering Conference (RE), 2012.

[SMAP-31] L. A. Hermoye, A. van Lamsweerde, and D. E. Perry, "A reuse based approach to security requirements engineering", Master Thesis Paper at University of Texas http://users.ece.utexas.edu /~perry/work/papers/060908-LH-reuse.pdf, 2006. Last access: November 2015.

[SMAP-32] L. A. Hermoye, A. van Lamsweerde, and D. E. Perry, "Attack patterns for security requirements engineering", Master Thesis Paper at University of Texas http://users.ece.utexas.edu /~perry/work/papers/060908-LH-threats.pdf, 2006. Last access: November 2015.

[SMAP-33] N. Heumesser, and F. Houdek, "Towards systematic recycling of systems requirement", in International Conference on Software Engineering (ICSE), 2003.

[SMAP-34] A. Hoffmann, T. Schulz, H. Hoffmann, S. Jandt, A. Roßnagel, and J. Leimeister, "Towards the Use of Software Requirement Patterns for Legal Requirements" in International Requirements Engineering Efficiency Workshop (REEW) at REFSQ, 2012.

[SMAP-35] A. Hoffmann, M. Shellner, H. Hoffmann, and J. M. Leimeister, "Requirement Patterns to Support Socio-Technical System Design", in Socio-technical Design of Ubiquitous Computing Systems Book, 2014.

[SMAP-36] A. Issa, and A. Al-Ali, "Use Case Patterns Driven Requirements Engineering", in International Conference on Computer Research and Development, 2010.

[SMAP-37] A.F. Jaramillo, "Non-functional requirements elicitation from business process models" in International Conference on Research Challenges in Information Science (RCIS), 2011.

[SMAP-38] J. Jensen, I. Tondel, M. Jaatun, P. Meland, and H. Andresen, "Reusable Security Requirements for Healthcare Applications" in International Conference on Availability, Reliability and Security (ARES), 2009.

[SMAP-39] W. Lam, "Developing component-based tools for requirements reuse: a process guide"in IEEE International Workshop on Software Technology and Engineering Practice, 1997.

[SMAP-40] W. Lam, "Achieving requirements reuse: A domain-specific approach from avionics", Journal of Systems and Software, 1997.

[SMAP-41] W. Lam, and S. Jones, "Mechanising requirements engineering: reuse and the application of domain analysis technology" in IEEE International Conference on Automated Software Engineering, 1997.

[SMAP-42] W. Liu, K. He, K. Zhang, and J. Wang, "Combining Domain-Driven Approach with Requirement Assets for Networked Software Requirements Elicitation" in IEEE International Conference on Semantic Computing, 2008.

[SMAP-43] O. Lopez, M. A. Laguna, and F. J. Garcia, "Metamodeling for Requirements Reuse", in Workshop em Engenharia de Requisitos (WER), 2002.

[SMAP-44] E. K. Karatas, B. Jydir, and A. Birturk, "Ontology-Based Software Requirements Reuse: Case Study in Fire Control Software Product Line Domain", in IEEE International Conference on Data Mining Workshop (ICDMW), 2014.

[SMAP-45] M. Kolp, J. Castro, and J. Mylopoulos, "Organizational Patterns for Early Requirements Analysis", in Conference on Advance Information Systems Engineering (CAISE), 2003.

[SMAP-46] S. Konrad, and B. H. C. Cheng, "Requirements patterns for embedded systems" in IEEE Joint International Conference on Requirements Engineering, 2002.

[SMAP-47] S. Konrad, and B. H. Cheng, "Real-time specification patterns" in International Conference on Software Engineering (ICSE), 2005.

[SMAP-48] R. J. Macasaet, M. Noguera, M. Rodriguez, J. L. Garrido, S. Supakkul, and L. Chung, "Micro-business behavior patterns associated with components in a requirements approach" in International Workshop on Experiences and Empirical Studies in Software Modelling, 2012.

[SMAP-49] A. Mahfouz, L. Barroca, R. Laney, and B. Nuseibeh, "Patterns for Service-Oriented Information Exchange Requirements", in Conference on Pattern Languages of Programs (PLoP), 2006.

[SMAP-50] M. Mannion, B. Keepence, and D. Harper, "Using viewpoints to define domain requirements", IEEE Software, 1998.

[SMAP-51] M. Mannion, B. Keepence, H. Kaindl, and J. Wheadon, "Reusing single system requirements from application family requirements" in IEEE International Conference on Software Engineering (ICSE), 1999.

[SMAP-52] M. Mannion, and H. Kaindl, "Using parameters and discriminants for product line requirements", Journal of Systems Engineering, 2008.

[SMAP-53] O. Mendez-Bonilla, X. Franch, and C. Quer, "Requirements Patterns for COTS Systems", in International Conference on Composition-Based Software Systems (ICCBSS), 2008.

[SMAP-54] A. Monzon, "A Practical Approach to Requirements Reuse in Product Families of On-Board Systems" in IEEE International Symposium on Requirements Engineering, 2008.

[SMAP-55] B. Moros, A. Toval, F. Rosique, and P. Sanchez, "Transforming and tracing reused requirements models to home automation models", Information and Software Technology, 2013.

[SMAP-56] T. Myklebust, N. Lyngby, R. Bains, and G. K. Hanssen, "CoVeR maintenance and maintainability requirements by using tools" in Reliability and Maintainability Symposium (RAMS), 2014.

[SMAP-57] T. Okubo, H. Kaiya, and N. Yoshioka, "Effective Security Impact Analysis with Patterns for Software Enhancement", in International Conference on Availability, Reliability and Security (ARES), 2011.

[SMAP-58] C. L. Pacheco, I. A. Garcia, J. A. Calvo-Manzano, and M. Arcilla, "A proposed model for reuse of software requirements in requirements catalog", Journal of Software: Evolution and Process, 2015.

[SMAP-59] C. Palomares, C. Quer, and X. Franch, "PABRE-Man: Management of a requirement patterns catalogue" in IEEE International Requirements Engineering Conference (RE), 2011.

[SMAP-60] C. Palomares, C. Quer, X. Franch, C. Guerlain, and S. Renault, "A catalogue of non-technical Requirement Patterns" in IEEE International Workshop on Requirements Patterns (RePa), 2012.

[SMAP-61] C. Palomares, C. Quer, X. Franch, C. Guerlain, and S. Renault, "A Catalogue of Functional Software Requirement Patterns for the Domain of Content Management Systems" in ACM Symposium on Applied Computing (SAC), 2013.

[SMAP-62] C. Palomares, C. Quer, and X. Franch, "PABRE-Proj: Applying patterns in requirements elicitation", in IEEE International Requirements Engineering Conference (RE), 2013.

[SMAP-63] M. C. Panis, "Reuse of Architecturally Derived Standards Requirements", in IEEE International Requirements Engineering Conference (RE), 2015.

[SMAP-64] S. Paydar, and M. Kahani, "A semantic web enabled approach to reuse functional requirements models in web engineering", Automated Software Engineering Journal, 2014.

[SMAP-65] S. Paydar, and M. Kahani, "A Semi-Automated Approach to Adapt Activity Diagrams for New Use Cases", Journal of Information and Software Technology, 2015.

[SMAP-66] C. Peper, R. Gotzhein, and M. Kronenburg, "A generic approach to the formal specification of requirements" in IEEE International Conference on Formal Engineering Methods, 1997.

[SMAP-67] C. Pichler, C. Huemer, and M. Strommer, "Evolution patterns for business document models", in International Software Product Line Conference, 2011.

[SMAP-68] A. Post, I. Menzel, A. Podelski, "Applying Restricted English Grammar on Automotive Requirements: Does it Work? A Case Study", in International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2011.

[SMAP-69] S. Renault, O. Mendez-Bonilla, X. Franch, C. Quer, "PABRE: Pattern-based Requirements Elicitation" in International Conference on Research Challenges in Information Science (RCIS), 2009.

[SMAP-70] M. Riaz, J. Slankas, J. King, L. Williams, "Using templates to elicit implied security requirements from functional requirements - a controlled experiment", in ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (SEM), 2014.

[SMAP-71] M. Riaz, J. King, J. Slankas, and L. Williams, "Hidden in plain sight: Automatically identifying security requirements from natural language artifacts" in IEEE International Requirements Engineering Conference (RE), 2014.

[SMAP-72] H. Roder, "A pattern approach to specifying usability features in use cases" in International Workshop on Pattern-Driven Engineering of Interactive Computing Systems, 2011.

[SMAP-73] H. Roder, "Specifying usability features with patterns and templates", in International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE), 2012.

[SMAP-74] K. Roher, and D. Richardson, "Sustainability Requirements Patterns", in International Workshop on Requirements Patterns (RePa), 2013.

[SMAP-75] M. Saeki, "Reusing use case descriptions for requirements specification: towards use case patterns" in Asia Pacific Software Engineering Conference (APSEC), 1999.

[SMAP-76] P. Salini, and S. Kanmani, "A knowledge-oriented approach to security requirements for an E-voting system", International Journal of Computer Applications, 2012.

[SMAP-77] E. Sarmiento, J. C. Sampaio do Prado Leite, and E. Almentero, "Using Correctness, Consistency, and Completeness Patterns for Automated Scenarios Verification", in IEEE International Workshop on Requirements Patterns (RePa), 2015.

[SMAP-78] P. Schnitzhofer, F. Schnitzhofer, and R. Ramler, "Tool Support for Reuse-Driven Elicitation and Specification of User Requirements", in EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), 2014.

[SMAP-79] L. Seungyun, K. Hyunmin, H. Myungjin, J. Dongcho, J. Junho, and K. Kangtae, "Reusable SW Requirements Development Process: Embedded SW Industry Experiences", in Australian Software Engineering Conference (ASWEC), 2007.

[SMAP-80] A. Shahrokni, and R. Feldt, "Towards a Framework for Specifying Software Robustness Requirements based on Patterns", in International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2010.

[SMAP-81] S. Silva, J. Araujo, A. Rodrigues, M. Urbieta, A. Moreira, S. Gordillo, and G. Rossi, "Reuse of spatial concerns based on aspectual requirements analysis patterns" in International Conference on Research Challenges in Information Science (RCIS), 2012.

[SMAP-82] G. Sindre, D. G. Firesmith, and A. L. Opdahl, "A reuse-based approach to determining security requirements", in International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ), 2003.

[SMAP-83] R. Slavin, J.-M. Lehker, N. Jianwei, and T.D. Breaux, "Managing security requirements patterns using feature diagram hierarchies", IEEE International Requirements Engineering Conference (RE), 2014.

[SMAP-84] S. Srivastava, "A Repository of Software Requirement Patterns for Online Examination System", International Journal of Computer Science, 2013.

[SMAP-85] M. Sultan, and A. Miranskyy, "Ordering Interrogative Questions for Effective Requirements Engineering: The W6H Pattern", in IEEE International Workshop on Requirements Patterns (RePa), 2015.

[SMAP-86] S. Supakkul, T. Hill, E. A. Oladimeji, and L. Chung, "Capturing, Organizing, and Reusing Knowledge of NFRs: An NFR Pattern Approach" in International Workshop on Managing Requirements Knowledge (MARK), 2009.

[SMAP-87] S. Supakkul, T. Hill, L. Chung, T. Tun, T. Thein, and J. C. S. do Prado Leite, "An NFR Pattern Approach to Dealing with NFRs", in IEEE International Requirements

Engineering Conference (RE), 2010.

[SMAP-88] S. Supakkul, and L. Chung, “Visualizing non-functional requirements patterns” in International Workshop on Requirements Engineering Visualization (REV), 2010.

[SMAP-89] E. Tambouris, E. Kaliva, M. Liaros, and K. Tarabanis, “A reference requirements set for public service provision enterprise architectures”, Software and Systems Modeling Journal, 2014.

[SMAP-90] S. F. Tjong, N. Hallam, M. Hartley, “Improving the Quality of Natural Language Requirements Specifications through Natural Language Requirements Patterns”, in International Conference on Computer and Information Technology (CIT), 2006.

[SMAP-91] A. D. Toro, “A Requirements Elicitation Approach Based in Templates and Patterns”, in Workshop em Engenharia de Requisitos (WER), 1999.

[SMAP-92] A. Toval, J. Nicolas, B. Moros, and F. Garcia, “Requirements Reuse for Improving Information Systems Security: A Practitioner Approach”, Requirements Engineering Journal, 2002.

[SMAP-93] A. Toval, J. M. Carrillo-de-Gea, J. L. Fernandez-Aleman, and R. Toval, “Learning systems development using reusable standard-based requirements catalogs” in IEEE Global Engineering Education Conference (EDUCON), 2011.

[SMAP-94] A. van Lamsweerde, “Elaborating security requirements by construction of intentional anti-models” in International Conference on Software Engineering (ICSE), 2004.

[SMAP-95] E. Vassev, and M. Hinchey, “Autonomy requirements engineering” in International Conference on Information Reuse and Integration (IRI), 2013.

[SMAP-96] E. Vassev, and M. Hinchey, “On the autonomy requirements for space missions” in IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2013.

[SMAP-97] C. Videira, and A. Rodrigues da Silva, “Patterns and metamodel for a natural- language-based requirements specification language”, in Conference on Advance Information Systems Engineering (CAISE), 2006.

[SMAP-98] R. S. Wahono, and J. Cheng, “Extensible requirements patterns of web application for efficient web application development”, in International Symposium on Cyber Worlds, 2002.

[SMAP-99] K. Watahiki, and M. Saeki, “Scenario patterns based on case grammar approach” in IEEE International Symposium on Requirements Engineering, 2001.

[SMAP-100] K. Watahiki, and M. Saeki, “Scenario evolution in requirements elicitation processes: scenario pattern and framework approach” in International Workshop on Principles of Software Evolution (IWPSE), 2001.

[SMAP-101] S. Withall, “Software Requirement Patterns”, Microsoft Press, 2007.

[SMAP-102] Y. Li, C. Pelties, M. Kaser, and N. Nararan, “Requirements patterns for seismology software applications”, in International Workshop on Requirements Patterns (RePa), 2012.

[SMAP-103] W. Yuhong, Z. Haihong, and L.Liu, "Analysing security requirements patterns based on problems decomposition and composition", in International Workshop on Requirements Patterns (RePa), 2011.

[SMAP-104] O. Zinovatna, and L. M. Cysneiros, "Reusing Knowledge on Delivering Privacy and Transparency Together", in IEEE International Workshop on Requirements Patterns (RePa), 2015.

Bibliography

[1] E. Hull, K. Jackson and J. Dick, Requirements Engineering (3rd Edition), Springer, 2011.

[2] Software Engineering Institute, CMMI for Development (CMMI-Dev), V1.2, 2006.

[3] International Standards Organization, ISO/IEC 15504-5 -- An Exemplar Process Assessment Model, 2007.

[4] M. Glinz, "On Non-Functional Requirements," in IEEE Internation Requirements Engineering Conference (RE), 2007.

[5] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," in International Conference on Software Engineering (ICSE), 2000.

[6] K. Pohl and C. Rupp, Requirements Engineering Fundamentals, O'Reilly, 2011.

[7] I. Jacobson, G. Booch and J. Rumbaugh, The unified software development process - the complete guide to the unified process from the original designers, Addison-Wesley Longman Publishing Co., 1999.

[8] J. Carvallo, X. Franch and C. Quer, "Managing Non-Technical Requirements in COTS Components Seelction," in IEEE International Requirements Engineering Conference (RE), 2006.

[9] S. Withall, Software requirement patterns, Microsoft Press, 2007.

[10] S. Supakkul, T. Hill, L. Chung, T. T. Tun and J. C. Sampaio do Prado Leite, "An NFR pattern approach to dealing with NFRs," in IEEE Internation Requirements Engineering Conference (RE), 2010.

[11] A. Hoffmann, M. Sollner, H. Hoffmann and J. M. Leimeister, "Towards trust-based software requirement patterns," in International Workshop on Requirement Patterns (REPA) at IEEE International Requirements Engineering Conference (RE), 2012.

[12] W. Lam, "A case-study of requirements reuse through product families," Annals of Software Engineering, vol. 5(1), pp. 253-277, 1998.

[13] K. Pohl, Requirements Engineering: Fundamentals, Principles, and Techniques. 1st edition, Springer, 2010.

[14] B. Lawrence, K. Weigers and C. Ebert, "The Top Risks of Requirements Engineering," IEEE Software, vol. 18, no. 6, 2001.

[15] J. D. Procaccino, J. M. Verner, S. P. Overmyer and M. E. Darter, "Case study: factors for early prediction of software development success," Information and software technology, vol. 44, no. 1, pp. 53-62, 2002.

[16] C. Jones, "Scoring and Evaluating Software Methods, Practices, and Results. Version 12.0," 2014. [Online]. Available: http://Namcookanalytics.com. [Accessed July 2015].

[17] T. Biggerstaff and A. Perlis, Software Reusability, Addison-Wesley, 1989.

[18] W. Schaefer, R. Prieto-Diaz and M. Matsumoto, Software Reusability, Ellis Horwood, 1994.

[19] E. Karlsson, Software Reuse: A Holistic Approach, John Wiley & Sons, 1995.

[20] I. Jacobson, M. Griss and P. Jonsson, Software Reuse: Architecture, Process and Organization for Business Success, Addison-Wesley, 1997.

[21] M. Ezran, M. Moricio and C. Tully, Practical Software Reuse, Springer, 2012.

[22] R. Prieto-Diaz, "Classification of Reusable Modules," Software Reusability, ACM press, 1989.

[23] N. Maiden and A. Sutcliffe, "Exploiting reusable specification through analogy," Communications of the ACM, vol. 35, no. 4, 1993.

[24] W. Lain, "Reasoning about requirements from past cases," 1994.

[25] M. Niazi, K. Cox and J. Verner, "A measurement framework for assessing the maturity of requirements engineering process," Software Quality Journal, vol. 16, no. 2, pp. 213-235, 2008.

[26] K. Wiegers and J. Beatty, Software Requirements, 3rd Edition, Microsoft Press, 2013.

[27] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King and S. Angel, A Pattern Language, Oxford University Press, 1977.

[28] D. Schmidt, M. Fayad and R. Johnson, Software Patterns, Communications of The ACM, 1996.

[29] E. Gamma, R. Helm, R. Johnson and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.

[30] F. Buschmann, R. Meunie, H. Rohnert, P. Sommerlad and M. Stal, Pattern- Oriented Software Architecture - A System of Patterns, John Wiley & Sons, 1996.

[31] M. Shaw, "Patterns for Software Architectures," in Pattern Languages of Program Design, Addison-Wesley, 1995.

[32] M. Fowler, Analysis Patterns: Reusable Object Models, Addison-Wesley, 1997.

[33] J. Coplien, "A Generative Development-Process Pattern Language," in Pattern Languages of Program Design, Addison-Wesley, 1995.

[34] B. Whitenack, "RAPPeL: A Requirements-Analysis-Process Pattern Language for Object-Oriented Development," in Pattern Languages of Program Design, Addison-Wesley, 1995.

[35] A. Cockburn, The Interaction of Social Issues and Software Architecture, Communications of The ACM, 1996.

[36] S. Ambler, Process Patterns: Building Large-Scale Systems Using Object Technology, Cambridge University Press, 1998.

[37] J. Kotula, “Using Patterns To Create Component Documentation,” IEEE Software, vol. 15, no. 2, 1998.

[38] B. Keepence and M. Mannion, “Using Patterns to Model Variability in Product Families,” IEEE Software, vol. 16, no. 4, 1999.

[39] E. Sadraei, A. Aurum, G. Beydoun and B. Paech, “A field study of the requirements engineering practice in Australian software industry,” Requirements Engineering, vol. 12, no. 3, pp. 145-162, 2007.

[40] K. Cox, M. Niazi and J. Verner, “Empirical study of Sommerville and Sawyer's requirements engineering practices,” IET software, vol. 3, no. 5, pp. 339-355, 2009.

[41] B. Solemon, S. Sahibuddin, Ghani and A. A. A., “Requirements engineering problems and practices in software companies: An industrial survey,” Advances in Software Engineering, 2009.

[42] SwissQ, “Requirements Trends & Benchmarks Switzerland 2012,” 2012. [Online]. Available: http://www.swissq.it/wp-content/uploads/2013/03/SwissQ_Req_Trends_2012_Web_EN.pdf. [Accessed February 2015].

[43] J. Schirpenbach, “Re-Use of Requirements via Libraries: Opportunities & Approaches,” Requirements Engineering Magazine, vol. 2, 2014.

[44] R. Young, “The Main Thing is Keeping the Main Thing the Main Thing,” Requirements Engineering Magazine, vol. 1, 2015.

[45] The Standish Group, “The Standish Group Report – Chaos,” 1995. [Online]. Available: http://www.projectsmart.co.uk/docs/chaos-report.pdf. [Accessed July 2015].

[46] T. Arnuphaptrairong, “Top ten lists of software project risks: Evidence from the literature survey,” in International Multi Conference of Engineers and Computer Scientists, 2011.

[47] D. Galorath, “Software Project Failure Costs Billions-Better Estimation & Planning Can Help,” 2012. [Online]. Available: http://galorath.com/wp/software-project-failure-costs-billions-better-estimation-planning-can-help/. [Accessed July 2015].

[48] Calleam Consulting, “Why do projects fail. 101 Common Causes,” 2015. [Online]. Available: http://calleam.com/WTPF/?page_id=2338. [Accessed July 2015].

[49] B. Boehm and V. Basili, “Software Defect Reduction Top 10 List,” Journal Computer, vol. 34, no. 1, p. 135–137, 2001.

[50] M. Eriksson, J. Börstler and K. Borg, “Managing requirements specifications for product lines–An approach and industry case study,” Journal of Systems and Software, vol. 82, no. 3, pp. 435-477, 2009.

[51] D. Hauksdottir, A. Vermehren and J. Savolainen, “Requirements reuse at Danfoss,” in IEEE International Requirements Engineering Conference (RE), 2012.

[52] Y. Li, C. Pelties, M. Kaser and N. Nararan, “Requirements patterns for seismology software applications,” in International Workshop on Requirements Patterns (REPA) at Requirements Engineering (RE)IEEE International Requirements Engineering Conference (RE), 2012.

[53] Y. Chernak, "Requirements reuse: the state of the practice," in International Conference on Software Science, Technology and Engineering (SWSTE), 2012.

[54] J. M. Carrillo-de-Gea, J. Nicolás, J. L. F. Alemán, A. Toval, A. Vizcaíno and C. Ebert, "Reusing requirements in global software engineering," in Managing requirements knowledge, Springer Berlin Heidelberg, 2012, pp. 171-197.

[55] A. Hoffmann, A. Janzen, H. Hoffmann and J. M. Leimeister, "Success Factors for Requirement Patterns Approaches Exploring Requirements Analysts' Opinions and Whishes," in Sozio-technisches Systemdesign im Zeitalter des Ubiquitous Computing (SUBICO), 2013.

[56] C. L. Pacheco, I. A. García, J. A. Calvo-Manzano and M. Arcilla, "A proposed model for reuse of software requirements in requirements catalog," Journal of Software: Evolution and Process, vol. 27, no. 1, pp. 1-21, 2015.

[57] L. Goldin and D. M. Berry, "Reuse of requirements reduced time to market at one industrial shop: a case study," Requirements Engineering, vol. 20, no. 1, pp. 23-44, 2015.

[58] F. B. V. Benitti and R. C. da Silva, "Evaluation of a Systematic Approach to Requirements Reuse," Journal of Universal Computer Science, vol. 19, no. 2, pp. 254-280, 2013.

[59] V. de Freitas Sodre, L. Jugurta, V. Vilela, M. Andrade and V. Marcus, "Improving productivity and quality of GIS databases design using an analysis pattern catalog," in Asia-Pacific Conference on Conceptual Modelling (APCCM), 2005.

[60] K. Beckers and M. Heisel, "A Foundation for Requirements Analysis of Privacy Preserving Software," in International Cross-Domain Conference and Workshop on Availability, Reliability, and Security (ARES), 2012.

[61] A. van Lamsweerde, "Elaborating security requirements by construction of intentional anti-models," in International Conference on Software Engineering (ICSE), 2004.

[62] A. Hoffmann, T. Schulz, H. Hoffmann, S. Jandt, A. Robnagel and J. Leimeister, "Towards the Use of Software Requirement Patterns for Legal Requirements," in International Requirements Engineering Efficiency Workshop (REEW) at International Working Conference on Requirements Engineering: Foundations for Software Quality (REFSQ), 2012.

[63] K. Haeng-Kon, "Effective Domain Modeling for Mobile Business AHMS (Adaptive Human Management Systems) requirements," in International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014.

[64] A. F. Jaramillo, "Non-functional requirements elicitation from business process models," in International Conference on Research Challenges in Information Science (RCIS), 2011.

[65] O. Lopez, M. A. Laguna and F. J. Garica, "Metamodeling for Requirements Reuse," in Workshop em Engenharia de Requisitos (WER), 2002.

[66] R. Wieringa, "Design science as nested problem solving," in International Conference on Design Science Research in Information Systems and Technology (DESRIST), 2009.

[67] M. Shaw, "The coming-of-age of software architecture research," in International Conference on Software Engineering (ICSE), 2001.

[68] GESSI group, Requirement-based production of service-oriented software, Research Project. GESSI group. Founded by Spanish government (TIN2010-19130-C02-00).

[69] GESSI group, Requirements Engineering for Multi-stakeholder Distributed Systems, Research Project. GESSI group. Acciones Integradas program (HU2005-0021). Founded by Spanish government..

[70] O. Mendez-Bonilla, X. Franch and C. Quer, "Requirements Patterns for COTS Systems," in International Conference on Composition-Based Software Systems (ICCBSS), 2008.

[71] S. Renault, O. Mendez-Bonilla, X. Franch and C. Quer, "PABRE: Pattern-based Requirements Elicitation," in International Conference on Research Challenges in Information Science (RCIS), 2009.

[72] S. Renault, O. Mendez-Bonilla, X. Franch and C. Quer, "A pattern-based method for building requirements documents in call-for-tender processes," International Journal of Computer Science & Applications (IJCSA), vol. 6, no. 5, pp. 175-202, 2009.

[73] C. Palomares, "Definition and Use of Software Requirement Patterns in Requirements Engineering Activities," in Doctoral Symposium at International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2014.

[74] K. Petersen, S. Vakkalanka and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," Information and Software Technology (IST), vol. 64, pp. 1-18, 2015.

[75] B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering," EBSE Technical Report EBSE-2007-01, 2007.

[76] C. Palomares, "Including Functional and Non-Technical Requirements in a Software Requirement Patterns Catalogue," Master Thesis at Universitat Politécnica de Catalunya (Spain), 2011.

[77] C. Palomares, "Definition and Use of Software requirement Patterns in Requirements Engineering Activities," Thesis Proposal at Universitat Politécnica de Catalunya (Spain), 2013.

[78] C. Palomares, C. Quer and X. Franch, "Online Questionnaire: Using a Pattern Catalogue in Requirements Engineering Activities," in Empirical Fair Track at International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2013.

[79] C. Palomares, C. Quer and X. Franch, "Requirements reuse and patterns: A Survey," in International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2014.

[80] C. Palomares, C. Quer and X. Franch, "Requirements Reuse and Requirement Patterns: A State of the Practice Survey," submitted to Empirical Software Engineering Journal, 2015.

[81] X. Franch, C. Palomares, C. Quer, S. Renault and F. Lazzer, "A Metamodel for Software Requirement Patterns," in International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2010.

[82] X. Franch, C. Quer, S. Renault, C. Guerlain and C. Palomares, "Constructing and Using Software Requirements Patterns," in Managing Requirements Knowledge, Springer, 2013.

[83] C. Palomares, C. Quer, X. Franch, S. Renault and C. Guerlain, "A Catalogue of Non-Technical Requirement Patterns," in International Workshop on Requirements Patterns (REPA) at IEEE International Requirements Engineering Conference (RE), 2012.

[84] C. Palomares, C. Quer, X. Franch, S. Renault and C. Guerlain, "A Catalogue of Functional Software Requirement Patterns for the Domain of Content Management Systems," in Symposium On Applied Computing (SAC), 2013.

[85] C. Palomares, C. Quer and X. Franch, "Requirements Reuse with the PABRE Framework," Requirements Engineering Magazine, vol. 2014, no. 01, 2014.

[86] C. Palomares, "Implementació del procés d'utilització de patrons de requisits," Bachelor Thesis at Universitat Politécnica de Catalunya (Spain). English translation: Implementation of the Process for Using Requirement Patterns , 2010.

[87] R. Yuste, "Aplicació de patrons de requeriments a projectes de selecció de software," Bachelor Thesis at Universitat Politécnica de Catalunya (Spain). English translation: Application of requirements patterns to software selection projects, 2011.

[88] C. Ramirez, "Millora de la integració de les eines que composen el sistema PABRE d'ajuda a la elicitació de requisits," Bachelor Thesis at Universitat Politécnica de Catalunya (Spain). English translation: Improvement of the integration of the tools that compose the PABRE system of help to the elicitation of requirements, 2011.

[89] H. Manrique, "Desenvolupament d'una interfície web per a l'aplicació PABRE-PROJ i adaptació de les seves funcions," Bachelor Thesis at Universitat Politécnica de Catalunya (Spain). English translation: Development of a web interface for the PABRE-PROJ tool and adaptation of its functionalities, 2012.

[90] J. Cubel, "Desenvolupament i integració d'un glossari de termes i registre de sinònims per a ser usat en les eines del sistema PABRE," Bachelor Thesis at Universitat Politécnica de Catalunya (Spain). English translation: Development and integration of a glossary and thesaurus to be used in the tools of the PABRE system, 2014.

[91] I. Lovric, "PABRE-Proj Interface Changes," Bachelor Thesis at Universitat Politécnica de Catalunya (Spain), 2014.

[92] F. Mora, "Development of web services for the PABRE system," Master Thesis at Universitat Politécnica de Catalunya (Spain), 2013.

[93] C. Palomares, C. Quer and X. Franch, "PABRE-Man: Management of a Requirement Patterns Catalogue," in Demo at IEEE International Requirements Engineering Conference (RE), 2011.

[94] C. Palomares, C. Quer and X. Franch, "PABRE-Proj: Applying Patterns in Requirements Elicitation," in Demo at IEEE International Requirements Engineering Conference (RE), 2013.

[95] X. Franch, C. Guerlain, C. Palomares, C. Quer and S. Renault, "Interested in Improving Your Requirements Engineering Process? Try Requirement Patterns!," in Empirical Fair Track at

International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2011.

[96] C. Alexander, The Timeless Way of Building, Oxford University Press, 1979.

[97] K. Petersen, R. Feldt, S. Mujtaba and M. Mattsson, "Systematic mapping studies in software engineering," in International conference on Evaluation and Assessment in Software Engineering (EASE), 2008.

[98] F. Braga de Azambuja, R. Melo Bastos and A. P. Terra Bacelo, "Systematic Review of Requirements Reuse," in International Conference on Software Engineering and Knowledge Engineering (SEKE), 2009.

[99] D. Mellado, C. Blanco, L. E. Sánchez and E. Fernández-Medina, "A systematic review of security requirements engineering," Computer Standards & Interfaces, vol. 32, no. 4, pp. 153-165, 2010.

[100] A. Souag, R. Mazo, C. Salinesi and I. Comyn-Wattiau, "Reusable knowledge in security requirements engineering: a systematic mapping study," Requirements Engineering, pp. 1-33, 2015.

[101] B. Kitchenham, P. Brereton, M. Turner, M. Niazi, S. Linkman, R. Pretorius and D. Budgen, "The impact of limited search procedures for systematic literature reviews A participant-observer case study," in Proceedings of the 3rd International Symposium on Empirical Software, 2009.

[102] B. Kitchenham, E. Mendes and G. Travassos, "Cross versus Within-Company Cost Estimation Studies: A Systematic Review," IEEE Transactions on Software Engineering, vol. 33, no. 5, pp. 316-329, 2007.

[103] M. Riaz, E. Mendes and E. Tempero, "A systematic review of software maintainability prediction and metrics," in Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement, 2009.

[104] J. Young and A. Antón, "A Method for Identifying Software Requirements Based on Policy Commitments," in IEEE International Requirements Engineering Conference (RE), 2010.

[105] C. Arora, M. Sabetzadeh, L. Briand, F. Zimmer and R. Gnaga, "Automatic Checking of Conformance to Requirement Boilerplates via Text Chunking: An Industrial Case Study," in International Conference on Evaluation and Assessment in Software Engineering (EASE), 2013.

[106] W. Lam, "Achieving requirements reuse: A domain-specific approach from avionics," Journal of Systems and Software, vol. 38, no. 3, pp. 197-209, 1997.

[107] N. Weston, R. Chitchyan and A. Rashid, "Formal semantic conflict detection in aspect-oriented requirement," in IEEE International Requirements Engineering Conference, 2008.

[108] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell and A. Wesslen, Experimentation in software engineering: an introduction, Springer, 2012.

[109] P. Mahendra and A. Ghazarian, "Pattens in Requirements Engineering: A Survey and Analysis Study," WSEAS Transactions on Information Science and Applications Journal, vol. 11, pp. 214-230, 2014.

[110] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: a systematic review," Information and Software Technology Journal, vol. 50, no. 9-10, pp. 833-859, 2008.

[111] W. Lam, J. MacDermid and A. Vickers, "Ten steps towards systematic requirements reuse," in IEEE International Requirements Engineering Conference (RE), 1997.

[112] D. C. Rine and N. Nada, "An empirical study of a software reuse reference model," Information and Software Technology, vol. 42, no. 1, pp. 47-65, 2000.

[113] A. Issa and A. Al-Ali, "Use Case Patterns Driven Requirements Engineering," in International Conference on Computer Research and Development (ICCRD), 2010.

[114] E. K. Karatas, B. Jydir and A. Birturk, "Ontology-Based Software Requirements Reuse: Case Study in Fire Control Software Product Line Domain," in IEEE International Conference on Data Mining Workshop (ICDMW), 2014.

[115] T. Myklebust, N. Lyngby, R. Bains and G. K. Hanssen, "CoVeR maintenance and maintainability requirements by using tools," in Reliability and Maintainability Symposium (RAMS), 2014.

[116] A. Toval, J. Nicolas, B. Moros and F. Garcia, "Requirements Reuse for Improving Information Systems Security: A Practitioner Approach," Requirements Engineering Journal, vol. 6, no. 4, pp. 205-219, 2002.

[117] T. Hall, S. Beecham and A. Rainer, "Requirements problems in twelve software companies: an empirical analysis," IEEE Proceedings, vol. 149, no. 5, pp. 153-160, 2002.

[118] C. J. Neill and P. A. Laplante, "Requirements engineering: the state of the practice," IEEE software, vol. 20, no. 6, pp. 40-45, 2003.

[119] J. Verner, K. Cox, S. Bleistein and N. Cerpa, "Requirements engineering and software project success: an industrial survey in Australia and the US," Australian Journal of Information Systems, vol. 13, no. 1, 2005.

[120] J. Iqbal, R. Ahmad, M. H. Nizam, M. Nasir and M. A. Noor, "Significant requirements engineering practices for software development outsourcing," in Australian Software Engineering Conference (ASWEC), 2013.

[121] S. Khankaew and S. Riddle, "A review of practice and problems in requirements engineering in small and medium software enterprises in Thailand," in IEEE International Workshop on Empirical Requirements Engineering (EmpiRE) at IEEE International Requirements Engineering Conference (RE), 2014.

[122] R. Matulevicius, "Survey of Requirements Engineering Practice in Lithuanian Software Development Companies," in Information Systems Development, 2005.

[123] M. Niazi, M. El-Atar, M. Usman and N. Ikram, "An empirical study identifying high perceived value requirements engineering practices in global software development projects," in International Conference on Software Engineering Advances (ICSEA), 2012.

[124] U. Nikula, J. Sajaniemi and H. Kälviäinen, "A State-of-the-practice Survey on Requirements Engineering in Small-and Medium-sized Enterprises," Research Report in Lappeenranta University

of Technology, Finland, 2000.

[125] D. Méndez and S. Wagner, "Naming the pain in requirements engineering: A design for a global family of surveys and first results from Germany," Information and Software Technology, vol. 57, pp. 616-643, 2015.

[126] B. Solemon, S. Sahibuddin and A. Ghani, "Adoption of requirements engineering practices in Malaysian software development companies," in International Conference on Advanced Software Engineering and Its Applications (ASEA), 2009.

[127] A. Tahir and R. Ahmad, "Requirement engineering practices-an empirical study," in International Conference on Computational Intelligence and Software Engineering (CiSE), 2010.

[128] P. Sawyer, I. Sommerville and S. Viller, "Requirements process improvement through the phased introduction of good practice," Software Process: Improvement and Practice, vol. 3, no. 1, pp. 19-34, 1997.

[129] I. Sommerville and P. Sawyer, Requirements engineering: a good practice guide, John Wiley & Sons, Inc., 1997.

[130] N. H. Bakar and Z. M. Kasirun, "Exploring software practitioners perceptions and experience in requirements reuse an empirical study in Malaysia," International Journal of Software Engineering and Technology, vol. 1, no. 2, pp. 33-42, 2014.

[131] D. A. Dillman, J. D. Smyth and L. M. Christian, Internet, phone, mail, and mixed-mode surveys: the tailored design method, John Wiley & Sons, 2014.

[132] International Standards Organization, ISO/IEC 25010 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, 2011.

[133] K. Wright, "Researching Internet-based populations: Advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services," Journal of Computer-Mediated Communication, vol. 10, no. 3, 2005.

[134] B. Kitchenham and S. L. Pfleeger, "Principles of survey research part 6: data analysis," ACM SIGSOFT Software Engineering Notes, vol. 28, no. 2, pp. 24-27, 2003.

[135] K. Krippendorff, Content analysis: An introduction to its methodology, 3rd edition, Sage Publications, 2012.

[136] N. Matloff, "From Algorithm to Z-Scores: Probabilistic and Statistical Modeling in Computer Science," 2009. [Online]. Available: http://heather.cs.ucdavis.edu/~matloff/probstatbook.html. [Accessed July 2015].

[137] International Standards Organization, ISO/IEC 9126-1-2001 Software Engineering Product Quality Part 1: Quality Model, 2001.

[138] International Standards Organization, IEEE 830 - IEEE Recommended Practice for Software Requirements Specifications, 1998.

[139] T. Dybå, "Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context," ACM SIGSOFT Software Engineering Notes, vol. 28, no. 5, pp. 148-157, 2003.

[140] A. Toval, J. M. Carrillo-de-Gea, J. L. Fernandez-Aleman and R. Toval, "Learning systems development using reusable standard-based requirements catalogs," in IEEE Global Engineering Education Conference (EDUCON), 2011.

[141] A. Shahrokni and R. Feldt, "Towards a framework for specifying software robustness requirements based on patterns," in International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), 2010.

[142] P. Filipovikj, M. Nyberg and G. Rodriguez-Navas, "Reassessing the pattern-based approach for formalizing requirements in the automotive domain," in IEEE International Requirements Engineering Conference (RE), 2014.

[143] J. Jensen, I. A. Tondel, M. G. Jaatun, P. H. Meland and H. Andresen, "Reusable Security Requirements for Healthcare Applications," in International Conference on Availability, Reliability and Security (ARES), 2009.

[144] S. Konrad and B. H. Cheng, "Requirements patterns for embedded systems," in IEEE International Requirements Engineering Conference (RE), 2012.

[145] T. Dybå, "An empirical investigation of the key factors for success in software process improvement," IEEE Transactions on Software Engineering, vol. 31, no. 5, pp. 410-424, 2005.

[146] J. R. Evans and A. Mathur, "The value of online surveys," Internet research, vol. 15, no. 2, pp. 195-219, 2005.

[147] W. M. Trochim and J. P. Donnelly, "Research methods knowledge base," 2006. [Online]. Available: http://www.socialresearchmethods.net/kb/. [Accessed July 2015].

[148] T. C. Lethbridge, S. E. Sim and J. Singer, "Studying software engineers: Data collection techniques for software field studies," Empirical Software Engineering, vol. 10, no. 3, pp. 311-341, 2005.

[149] A. E. Milewski, "Global and task effects in information-seeking among software engineers," Empirical Software Engineering, vol. 12, no. 3, pp. 311-326, 2007.

[150] M. Umarji and S. E. Sim, "Archetypal Internet-Scale Source Code Searching," inding Source Code on the Web for Remix and Reuse, pp. 35-52, 2013.

[151] A. Solinski and K. Petersen, "Prioritizing agile benefits and limitations in relation to practice usage," Software Quality Journal, pp. 1-36, 2014.

[152] H. Boone and D. Boone, "Analyzing Likert Data," Journal of Extension, vol. 50, no. 2, 2012.

[153] S. Jamieson, "Likert scales: how to (ab) use them," Medical education, vol. 38, no. 12, pp. 1217-1218, 2004.

[154] R. R. Young, The Requirements Engineering Handbook, Artech House, 2004.

[155] S. Robertson and J. Robertson, Mastering the Requirements Process, 3rd Edition, Addison-Wesley, 2013.

[156] International Standards Organization, ISO/IEC/IEEE 29148 - Systems and software engineering — Life cycle processes — Requirements engineering, 2011.

[157] J. Corbin and A. Strauss, "Grounded Theory Research: Procedures, Canons, and Evaluative Criteria," Qualitative Sociology Journal, vol. 13, no. 1, pp. 3-21, 1990.

[158] B. S. Ali and Z. M. Kasirun, "A Review on Approaches for Identifying Crosscutting Concerns," in International Conference on Advanced Computer Theory and Engineering (ICACTE), 2008.

[159] L. de Alencar Sobral and L. Fernandes da Silva, "GT4CCI: An Approach Based on Grounded Theory for Crosscutting Concerns Identification in Requirements Documents," in Workshop em Engenharia de Requisitos (WER), 2013.

[160] International Standards Organization, IEEE Standard 610.12: Glossary of Software Engineering Terminology, 1990.

[161] K. Chen, W. Zhang, H. Zhao and H. Mei, "An Approach to Constructing Feature Models Based on Requirements Clustering," in IEEE International Conference on Requirements Engineering (RE), 2005.

[162] C. Duan, P. Laurent and J. Cleland-Huang, "Towards automated requirements prioritization and triage," Requirements Engineering Journal, vol. 14, no. 2, pp. 73-89, 2009.

[163] W. Frakes, R. Prieto-Diaz and C. Fox, "DARE: Domain analysis and reuse environment," Annals of Software Engineering Journal, vol. 5, no. 1, pp. 125-141, 1998.

[164] E. Yu, "Modelling Strategic Relationships for Process Engineering," PhD Thesis at University of Toronto, 1995.

[165] G. Mussbacher, J. Whittle and D. Amyot, "Semantic-Based Interaction Detection in Aspect-Oriented Scenarios," in IEEE International Requirements Engineering Conference (RE), 2009.

[166] M. Mannion, B. Keepence, H. Kaindl and J. Wheadon, "Reusing single system requirements from application family requirements," in International Conference on Software Engineering (ICSE), 1999.

[167] R. Smith, G. Avrunin, L. Clarke and L. Osterwell, "PROPEL: an approach supporting property elucidation," in International Conference on Software Engineering (ICSE), 2002.

[168] B. Moros, C. Vicente-Chicote and A. Toval, "Metamodeling Variability to Enable Requirements Reuse," in International Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD), 2008.

[169] H. Gomaa, "Reusable Software Requirements and Architectures for Families of Systems," Journal of Systems and Software, vol. 28, no. 1, pp. 189-202, 1995.

[170] L. Chung, B. Nixon, E. Yu and J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers, 2000.

[171] X. Franch and J. Carvallo, "Using Quality Models in Software Package Selection," IEEE Software, vol. 20, no. 1, pp. 34-41, 2003.

[172] A. Egyed and P. Grünbacher, "Identifying Requirements Conflicts and Cooperation: How Quality Attributes and Automated Traceability Can Help," IEEE Software, vol. 21, no. 6, pp. 50-58, 2004.

[173] C. J. Davis, A. C. Gillies, P. Smith and J. B. Thompson, "Current practice in software quality and the impact of certification schemes.," Software Quality Journal, vol. 2, no. 3, pp. 145-161, 1993.

[174] J. Cleland-Huang, R. Settimi, Z. Xuchang and P. Solc, "The Detection and Classification of Non-Functional Requirements with Application to Early Aspects," in IEEE International Requirements Engineering Conference (RE), 2006.

[175] N. M. Minhas, S. Majeed, Z. Qayyum and M. Aasem, "Controlled vocabulary based software requirements classification," in Malaysian Conference in Software Engineering (MySEC), 2011.

[176] C. Rolland and N. Prakash, "Matching ERP System Functionality to Customer Requirements," IEEE Computer, pp. 66-75, 2001.

[177] X. Franch, C. Quer, J. Cantón and R. Salietti, "Experience Report on the Construction of Quality Models for Some Content Management Software Domain," in International Conference on Composition-Based Software Systems (ICCBSS), 2008.

[178] B. Barns, W. D. U. Nat. Sci. Found. and T. Bollinger, "Making Reuse Cost-Effective," IEEE Software, vol. 8, no. 1, pp. 13-24, 1991.

[179] J. Poulin, Measuring Software Reuse: Principles, Practices and Economic Models, Addison Wesley, 1997.

[180] M. Daneva, "Understanding Functional Reuse of ERP Requirements in the Telecommunication Sector," in International Workshop on Softwrae Management (IWMS) at the International Conference on Software Process and Product Measurement (SPPM), 2008.

[181] J. Poulin, "Determining the Value of a Corporate Reuse Program," in IEEE Computer Society International Software Metrics Symposium (METRICS), 1993.

[182] A. MacCormack, J. Rusnak and C. Baldwin, "Exploring the duality between product and organizational architectures," Harvard Business School Research Paper, 2011.

[183] C. Baldwin and K. Clark, Design Rules: The Power of Modularity, MIT Press, 1999.

[184] J. Carriere, R. Kazman and I. Ozkaya, "A cost-benefit framework for making architectural decisions in a business context," in Inernational COnference on Software Engineering (ICSE), 2000.

[185] K. Sullivan, W. Griswold, Y. Cai and B. Hallen, "The structure and value of modularity in software design," in Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE), 2001.

[186] T. Bollinger and S. Pfleeger, "The economics of software reuse: issues and alternatives," Information and Software Technology, 1990.

[187] D. Nazareth and M. Rothenger, “Assessing the cost-effectiveness of software reuse: A model for planned reuse,” Journal of Systems and Software, vol. 73, no. 2, pp. 245-255, 2004.

[188] M. Rothenberger and K. Dooley, “A Performance Measure for Software Reuse Projects,” Decision Sciences Journal, vol. 30, no. 4, pp. 1131-1153, 1999.

[189] W. Frakes and C. Terry, “Software Reuse: Metrics and Models,” ACM Computing Surveys, vol. 28, no. 2, pp. 415-435, 1996.

[190] A. Mili, S. Chmiel, R. Gottumukkala and L. Zhang, “An integrated cost model for software reuse,” in International Conference on Software Engineering (ICSE), 2000.

[191] M. Ali, M. Ali Babar and K. Schmid, “A Comparative Survey of Economic Models for Software Product Lines,” in Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2009.

[192] P. Clements, J. McGregor and S. Cohen, “The Structured Intuitive Model for Product Line Economics (SIMPLE),” Technical Report CMU/SEI, 2005.

[193] J. Poulin, “The economics of product line development,” International Journal of Applied Software Technology, vol. 3, 1997.

[194] J. Letouzey, “The SQALE Method for Evaluating Technical Debt,” in International Workshop on Managing Technical Debt (MTD) at International Conference in Software Engineering (ICSE), 2012.

[195] J. Gaffney and R. D. Cruickshank, “A general economics model of software reuse,” in International Conference on Software Engineering (ICSE), 1992.

[196] H. Erdogmus, J. Favaro and W. Strigel, “Guest Editors' Introduction: Return on Investment,” IEEE Software, vol. 21, no. 3, pp. 18-22, 2004.

[197] R. Van Solingen, “Measuring the ROI of software process improvement,” IEEE Software, vol. 21, no. 3, pp. 32-38, 2004.

[198] S. Biffl, A. Aurum, B. Boehm, H. Erdogmus and P. Grünbacher, Value-Based Software Engineering, Springer-Verlag, 2005.

[199] G. Böckle, P. Clements, J. McGregor, D. Muthig and K. Schmid, “Calculating ROI for Software Product Lines,” IEEE Software, vol. 21, no. 3, pp. 23-31, 2004.

[200] R. Denney, “ROI on Requirements Management Tools,” 2003. [Online]. Available: http://www.agileconnection.com/article/calculating-roi-your-investment-requirements-management-tools. [Accessed July 2015].

[201] R. Moksony, “ROI elements of requirements management tools,” 2014. [Online]. Available: http://intland.com/blog/requirements/roi-elements-of-requirements-management-tools/. [Accessed July 2105].

[202] S. Decker, H. Cummins Inc. and J. Dager, “Software Product Lines Beyond Software Development,” in International Software Product Line Conference (SPLC), 2007.

[203] D. Ganesan, D. Muthig and K. Yoshimura, "Predicting return-on-investment for product line generations," in International Software Product Line Conference (SPLC), 2006.

[204] R. Van Solingen, "Measuring the ROI of software process improvement," IEEE Software, vol. 21, no. 3, pp. 32-38, 2004.

[205] M. Krystkowiak, V. Bétry and E. Dubois, "Efficient COTS Selection with OPAL Tool," in International Workshop on Models and Processes for the Evaluation of COTS Components (MPEC), 2004.

[206] J. J. López-Fernández, E. Guerra and J. de Lara, "Assessing the quality of meta-models," in Workshop on Model Driven Engineering,Verification and Validation (MoDeVVa), 2014.

[207] M. F. Bertoa and A. Vallecillo, "Quality attributes for software metamodels," in TOOLS Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE), 2010.

[208] G. Hinkel, M. Kramer, E. Burger, M. Strittmatter and L. Happe, "An Empirical Study on the Perception of Metamodel Quality," in International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 2016.

[209] H. Erdogmus, "How important is evidence really?," IEEE Software, vol. 27, no. 3, pp. 2-5, 2010.

[210] B. A. Kitchenham, T. Dybå and M. Jorgensen, "Evidence-based software engineering," in International Conference on Software Engineering (ICSE), 2004.

[211] C. Robson, Real World Research: A Resource for Social Scientists and Practitioner-researchers, Blackwell Publishers Inc, 2002.

[212] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," Empirical Software Engineering, vol. 14, no. 2, pp. 131-164, 2009.

[213] D. Méndez, S. Wagner, K. Lochmann, A. Baumann and H. de Carne, "Field study on requirements engineering: Investigation of artefacts, project parameters, and execution strategies," Information & Software Technology, vol. 6, no. 576-587, p. 49, 2012.

[214] B. J. Oates, Researching Information Systems and Computing, Sage Publications, 2006.

[215] W. G. Lutters and C. Seaman, "Revealing actual documentation usage in software maintenance," Information and Software Technology, vol. 49, no. 6, pp. 576-587, 2007.

[216] M. Q. Patton, Qualitative Research & Evaluation Methods, Sage Publications, 2002.

[217] J. Saldana, The coding manual for qualitative research, SAGE Publications, 2009.

[218] T. Dybå, D. I. K. Sjoberg and D. S. Cruzes, "What works for whom, where, when and why?: on the role of context in empirical software engineering," in ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2012.

This thesis proposes the use of Software Requirement Patterns (SRPs) as a means to capture and reuse requirements knowledge in the context of information technology projects in order to obtain requirements specifications of higher quality with a more effective requirements engineering process. Following the typical context-problem-solution structure of patterns, an SRP mainly consists of: a template (solution) that may generate one or more requirements when applied in a certain project, and some information (context-problem) to identify its applicability in that project. To facilitate their use, SRPs are encapsulated inside the PABRE (PAttern-Based Requirements Elicitation) framework.

The framework covers all the elements that could be critical for the adoption of a requirements reuse technique. Specifically, the framework includes: 1) A metamodel that describes the structure of SRPs and their organization inside a catalogue; 2) An SRP catalogue composed by non-functional, non-technical and functional SRPs, the last ones being specific for the content management system domain; 3) A method for guiding the use of an SRP catalogue during requirements elicitation and specification, as well as another one for constructing and updating it; 4) An economic model to perform cost-benefit analysis on the adoption of SRPs based on return-on-investment; and 5) The PABRE system as technological support.