UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Open source software ecosystems quality analysis from data sources

by

Oscar Hernán Franco Bedoya

Open Source Software Ecosystems Quality Analysis from Data Sources

Presented by:

Oscar Hernán Franco Bedoya

Advisors:

Dr. David Ameller and Dr. Dolors Costal

PhD thesis

Submitted in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy in Computing

UNIVERSITAT POLITÈCNICA de

C A T A L U N Y A

BARCELONA

Departament de Ciències de la Computació

December 2017

Acknowledgments

I remember that several years ago, I wanted a PhD in software engineering. However, the rector of the university where I worked told me that I was too old for a PhD (I was 37 years old). Then, I decided that I had to undertake a PhD. This has been a wonderful journey with laughter and tears, but above all with a lot of learning and new and good friends (It is remarkable after 40).

This research project would not have been possible without the support of many people who have been there for me for the last five years. First and foremost, I would like to thank my wife and partner **Claudia**, she was always there through the ups and downs of this process. My Family is next, my son **Thiago,** my mother **Bertha**, my sister **Miti**, **Camila** my happy nephew and my brother in law **John**. Thank you for all your support despite the time and distance. Also, I specifically have to thank my mother and mother in law **Nelida**, for traveling twice to take care of Thiago.

I would like to express my sincerest thanks to my advisors, Professors **David Ameller** and **Dolors Costal** for giving me the opportunity, the support and the guidance I needed. Their know-how, discipline and the quality of their work was inspiring and helped me to focus in my project. Thanks to professor **Xavi Franch** too, who gave me the opportunity to research in software engineering. He laid the first stone of this project.

My colleagues in the office, **Oscar Cabrera**, **Marc Oriol**, **Lidia López**, **Xavier Oriol**, **Silverio Martínez**, Lucia Mendez, **Edith Zavala**, **Cristina Palomares**, **Nikolas Galanis** and all the GESSI members, who have helped me through all these years, and with whom I have shared many moments.

I would like to thank to the **PhD committee, anonymous reviewers** of the thesis and papers published during the PhD, who provided priceless feedback and great ideas, so I could make of this thesis a much better version.

Last, but not least, a very special thanks to my friends, **Luis Castillo**, **Gustavo Isaza** and **Jhon Candamil** for keeping in touch with me despite the time and distance.

Abstract

Background: Open source software (OSS) and software ecosystems (SECOs) are two consolidated research areas in software engineering. The adoption of OSS by firms, governments, researchers and practitioners has been increasing rapidly in the last decades, and in consequence, they find themselves in a new kind of ecosystem composed by software communities, foundations, developers and partners, namely Open Source Software Ecosystem (OSSECO). In order to perform a systematic quality evaluation of a SECO, it is necessary to define certain types of concrete elements. This means that measures and evaluations should be described (e.g., through thresholds or expert judgment). The quality evaluation of an OSSECO may serve several purposes, for example: adopters of the products of the OSSECO may want to know about the liveliness of the OSSECO (e.g., recent updates); software developers may want to know about the activeness (e.g., how many collaborators are involved and how active they are); and the OSSECO community itself to know about the OSSECO health (e.g., evolving in the right direction). However, the current approaches for evaluating software quality (even those specific for open source software) do not cover all the aspects relevant in an OSSECO from an ecosystem perspective.

Goal: The main goal of this PhD thesis is to support the OSSECO quality evaluation by designing a framework that supports the quality evaluation of OSSECOs.

Methods: To accomplish this goal, we have used and approach based on design science methodology by Wieringa [1] and the characterization of software engineering proposed by M. Shaw [2], in order to produce a set of artefacts to contribute in the quality evaluation of OSSECOs and to learn about the effects of using these artefacts in practice.

Results: We have conducted a systematic mapping to characterize OSSECOs and designed the QuESo framework (a framework to evaluate the OSSECO quality) composed by three artifacts: (i) QuESo-model, a quality model for OSSECOs; (ii) QuESo-process, a process for conducting OSSECO quality evaluations using the QuESo-model; and (iii) QuESo-tool, a software component to support semi-automatic quality evaluation of OSSECOs. Furthermore, this framework has been validated with a case study on Eclipse.

Conclusions: This thesis has contributed to increase the knowledge and understanding of OSSECOs, and to support the quality evaluation of OSSECOs.

Keywords: software ecosystem, open source software, open source software ecosystem, open source software quality model, quality assessment, Eclipse, Bayesian network, quality assessment process, quality assessment tool research design.

Table of contents

1 Introduction

"Books are mirrors: you only see in them what you already have inside you."
— **Carlos Ruiz Zafón**, **The Shadow of the Wind**

1.1 Context and terminology

The present thesis has grown around two consolidated research areas in software engineering: Open Source Software (OSS) and Software Ecosystems (SECOs). The first area influences the way organizations develop, acquire, use and commercialize software. The second area have emerged as a paradigm to understand dynamics and heterogeneity in collaborative software development. For this reason, SECOs appear as a valid instrument to understand the relationships among OSS heterogeneous elements. Therefore, the context of this thesis is a combination of the OSS and SECO topics, i.e., OSS Ecosystems (OSSECOs).

OSS, SECO and OSSECO concepts are described in the following subsections in order to contextualize the work presented in this thesis.

1.1.1 Open Source Software

Nowadays, the adoption of OSS by organizations has become a strategic issue in a wide variety of application areas. As a consequence, organizations are increasingly becoming OSS adopters, and also several strategies for OSS adoption have already been identified and analyzed (usually followed by the industry) [3].

The OSS development approach has assisted in the spread of emerging technologies, allowing users to utilize freely publicly available software and developers to incorporate third party source code into their implementations. Individual and already tested libraries are often used as building blocks for larger software systems, offering reusable functionality and providing the means for faster time-to-release [4].

> “*The **OSS** concept covers software artifacts including code, licenses, development best practices, promotion, diffusion, innovation, social justice, ethics, philosophy, social movement, community, culture, governance, and organizational engagement.*” [204].

1.1.1.1 OSS Projects and Communities

OSS projects are typically initiated by an individual or a small group with a specific need. This need is the motivation for the creation of the OSS project [5]. Riehle [6], argue that: *“rather than a single corporate entity owning the software, a sometimes-broad community of volunteers determines which contributions are accepted into the source code base and where the OSS project is heading*” (e.g., R OSSECO [7]).

OSS communities are keystone actors of OSS projects. They guarantee the development, support and maintenance of OSS [8]. An OSS community involves organizations and individuals producing and consuming OSS components. There are many roles in an OSS community with different levels of participation e.g., users, reviewers, contributors, administrators, partners and developers [9] (e.g., Eclipse OSSECO). As noted by several authors such as Uden et al., [5], “the OSS community forms a particular and complex system endowed by an inner short timescale dynamic and a long timescale evolutionary dynamic”. OSS communities need to retain and attract their members in order to capture user innovation and to maintain the quality of the OSS projects [10]. Typically, OSS communities surrounding OSS projects provide access to all of the data related to their evolution. These can be used to evaluate the quality of OSSECOs.

1.1.1.2 OSS Resources

OSS projects accelerate and support the adoption of emerging component-based collaborative platforms. OSS projects typically provide access to several kinds of data sources to extract information about their evolution and the symbiotic relationships between OSS actors [11], [12]. According to Buford et al., [13] collaboration tools help OSS communities to coordinate their activities and enable groups of adopters and providers to work as a team, sharing information and communicating as needed, without being co-located. Some of the traditional OSS data sources are: version control systems, mailing lists, bug trackers, web sites, wikis, discussion forums, etc. However, there are also non-traditional data sources such as adopter feedback, market share reports, sales reports, OSS actor surveys, decision-making notes, expert surveys, etc.

1.1.1.3 OSS Analysis

OSS projects typically provide public availability of historical data, which facilitates the analysis of OSS evolution [14]. A number of studies have investigated large, well-known OSS projects through quantitative analysis (e.g., Linux kernel, Apache, Mozilla, Gnome, KDE). Several of these studies focus on social network analysis [15]. They rely on repository mining techniques to extract relevant data from OSS repositories or other data sources frequently used by OSS communities. These works use empirical software engineering methods by exploring and studying the OSS communities, including the way they work, cooperate, communicate and share information [16].

The interest in the research on software repositories is increasing, in particular the software repository mining community focuses on the analyses of the data available in OSS repositories. Xie et al, [17] explain that *"the researchers of this community explore a range of software engineering questions such as: software evolution, models of software development processes, characterization of developers and their activities, prediction of future software qualities, use of machine learning techniques on software project data, software bug prediction, analysis of software change patterns, and analysis of code clone"*. However, obtaining data from OSS repositories is a tedious exercise, and the obtained datasets are often non-homogeneous, which makes further analysis difficult [18]. However, there are collaborative development sites like GitHub that provide access to their internal data stores through an extensive REST application programming interface, which enables researchers to identify a rich collection of OSSECO information (e.g., OSSECO cross-references, technical dependencies between projects).

1.1.2 Software Ecosystems

The term software ecosystem was coined by Messerschmitt & Szyperski, [19]. They defined a SECO as: "a collection of software products that have some given degree of symbiotic relationships".

SECOs have emerged in the last years as a novel way to understand the relationships between software projects, products, communities, and organizations [20]. Furthermore, they are increasingly popular because of their economic, strategic, and technical advantages [21]. Unfortunately, in contrast to natural

- **Business view:** " *we define a software ecosystem as a set of businesses functioning as a unit and interacting with a shared market for software and services* " [25].
- **Platform view:** " *a software ecosystem is a collection of software projects which are developed and evolve together in the same environment* " [27].

ecosystems, there is not a common definition of SECO. A SECO can be defined and interpreted in different ways, depending on the point of view [22]. Two main viewpoints for SECO can be identified, namely business centric and platform-centric:

The first view emphasizes a holistic business-oriented perspective of a SECO as a network of actors, organizations and companies. It is adopted by authors as Messerschmitt & Szyperski [19], Bosch [23], and Jansen, et al., [24]. This view is similar to the commercial software ecosystem category in Bosch [23], the external view level in Jansen et al., [25], the business dimension in dos Santos & Werner [26], and the ecosystem-in-the-large in Goeminne & Mens [12].

The second view highlights technical and social aspects of a set of software projects, technical platforms and communities. This perspective is adopted by authors such as Lungu [27] and Mathieu Goeminne & Mens [28]. This view is similar to the social software ecosystems in Bosch [23], the internal view level in Jansen et al., [24], the social and architectural dimensions in dos Santos & Werner [26], and the ecosystem-in-the small in Goeminne & Mens, (2013) [12]. Platform-centric viewpoint focuses on the platform environment.

1.1.3 Open Source Software Ecosystems

Similarly to what happens with SECOs, OSSECOs can also be understood from different perspectives. (1) an ecosystem perspective, where OSSECOs are a network of actors, organizations and companies with symbiotic relationships that can be studied from a business-goal point of view; (2) a project-community perspective that focuses on technical and social aspects of a set of software projects and their communities [16], [29]. Some authors argue that OSSECOs are probably the most complex type of SECO [30]. However, there are very few definitions of OSSECO in the literature.

- *"An arrangement of individual and organizational units, involved in or affecting the circulation, transformation, and accumulation of capital (in various forms) in order to provide cooperative development, testing, marketing, distribution, implementation, and support of open source software"* [31].
- *"An OSS ecosystem is one where it is possible to add contributions to a project, create and publish components in the extension market, etc., without any barriers*" [32].
- *"A set of developers functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. The result of the interaction is freely available for everyone"* [33].
- *"A SECO placed in a heterogeneous environment, whose boundary is a set of niche players and whose keystone player is an OSS community around a set of projects in an open-common platform"* [34].

1.2 Motivation

OSSECOs is a growing research area in software engineering. A number of studies have investigated large, well-known OSSECOs through quantitative analysis (e.g., Linux kernel, Apache, Mozilla, Gnome, KDE). In the last decade, the study of OSS has changed significantly because of OSS employs new types of: socialization processes, development practices, community networking, business models, organization structure, governance and legality [35]. On the other hand, SECOs are increasingly popular for their economic, strategic, and technical advantages [21] and, also as Wasserman argue: "*there is a growing trend of companies contributing to FOSS projects. In some cases, companies contribute people and/or code to projects as a way of supporting the project"* [36]. Furthermore, SECOs provide a new holistic point of view for understanding OSS. In this sense, OSS initiatives typically create an adequate environment for making a SECO emerge from their projects, communities and external actors (partners, public and private institutions, research groups). This means that SECOs provide a practical approach to understand all the synaptic relationships between OSS heterogeneous elements. This thesis aims at uncovering the existing research on OSS ecosystems (OSSECOs).

To date however, there are no studies focused specifically on OSSECOs. In fact, several secondary studies explicitly mention the need for further study in the field of OSSECOs. The first study in that direction was Jansen et al., [24] who mentioned several challenges, remarkably characterization and modelling of SECOs. Barbosa et al., [37] identified eight major fields within the software ecosystems domain; it is worth mentioning that one of them is the further study of OSSECOs. Hanssen & Dybå [38] uncovered several theoretical challenges about SECO, particularly related to socio-technical theory. Finally, Manikas [39] proposed two approaches to address the complexity and the theory building in SECOs. Table 1 summarizes these SECO challenges.

Modeling and quality are two of the OSSECO challenges identify by [24] and [37] respectively. Adewumi et al., [40] classified OSS quality models into two generations: the first generation covers the years 2003-2006. These models were the initial attempts to evaluate the OSS quality (i.e., OSMM [41], QSOS [42], OBRR [43] and Sung et al. model [44]), the second generation covers from 2007 to 2015, it include the quality models: QualOSS [45], OMM [46], SQO-OSS [47] and EFFORT [48]. Adewumi et al., concluded that the key difference between the two generations of models is that the second-generation models provide more tools to aid the quality evaluation process compared to the first-generation models. However, none of the above OSS quality models addressed quality from a software ecosystem point of view.

In this thesis, we have identified three different stakeholders in which an actor wants to know the quality of an OSSECO: (1) potential adopters: organizations or individuals who are interested in using the OSSECO projects. (2) potential contributors: individuals who are interested in some form of collaboration (e.g., coding, documenting, reporting bugs) with the OSSECO projects. Finally, (3) OSSECO members: authors, current adopters, passive users, etc., who are interested in the OSSECO quality assurance.

In conclusion, the motivation behind this thesis is the need of a complete framework to evaluate the quality of OSSECOs that supports current and potential ecosystem members, as well as researchers in the OSSECO analysis process.

Table 1 SECO challenges, (highlighted the ones related with OSSECOs).

Authors	Challenge
[24]	**Characterization and modelling of SECOs** ○ Developing Policies and strategies within SECOs for SECO orchestration ○ Determining a strategy to thrive and make profit in an SSN
[37]	**How quality can be measured per develop** ○ How relationships are formed between developers **How conflicts are solved in OSS ecosystems** ○ How decisions are made in SECO and how can be measured in code changes ○ How APIs to third-party component are used
[38]	**Socio-technical theory** **Related theory of organizational ecology**
[39]	**Software ecosystem scoping** ○ Theory building

1.3 Research goals and research questions

Based on the current research challenges for SECOs and in a research roadmap for OSSECOs presented in a previous work [34], the research gaps in which we aim to contribute are threefold:

- **OSSECO state of the art:** According to several authors, such as: Barbosa et al. [37], Manikas [39], Axelsson & Skoglund [49], there is a lack of secondary studies in OSSECOs. In addition, OSSECOs are important objects of study because they are SECOs with specific characteristics, boundaries, actors and so on. Therefore, conducting a state of the art of OSSECOs is justified.

- **OSSECO quality:** The quality of OSSECOs affects organizations, adopters, software developers (i.e., OSSECO stakeholders) and the OSSECO itself. However, quality management and operationalization of software ecosystems is still an immature discipline. The ISO standard quality model (ISO/IEC-25010, 2010) is not enough for characterizing the quality of OSSECOs in terms of production process, community, distribution methods, social organization, support, etc. [48]. Because of this, OSS quality models emerged due to the inability of traditional quality models to measure this OSS unique features [40]. These quality models on OSS projects can be the basis of OSSECO quality models [50]. These models should ensure that the OSSECO quality evaluation covers the most important quality characteristics from an ecosystem point of view.
- **OSSECO monitoring:** The assessment of OSSECO measures is usually realized by tools for a particular community or specific platform. For instance, there exist several solutions in the literature for the monitoring and analysis of specific OSS communities by accessing directly to their available data repositories [51], [15], [52]. There is a need for implementing frameworks able to: (i) monitor a list of OSSECO quality measures along time (ii) provide a portfolio of web services that supports OSSECO monitoring.

1.3.1 Research goals

The goals of this thesis are defined to cope with the research gap previously specified, they are formulated in terms of the following Design Problem (DP):

> ***DP:*** *Identify the* ***knowledge about OSSECO*** *and based on it, provide the* ***QuESo-framework*** *in order to support the quality evaluation of OSSECOs****.***

- **To investigate the state of the art on OSSECOs.** The research question to achieve this objective is formulated in RQ1 (see Table 2). Since this main research question is very general, we refined it into finer-grained sub-questions. First, we want to identify the proposals in the field, find their evolution along the time, the information about geographical distribution of the publications and the classification between academy and industry, in summary, the demographic characterization of the studies (RQ1.1). Second, we want get information about: the most relevant definitions and general characteristics of OSSECOs (RQ1.2). Third, we want to how the knowledge about OSSECO is represented (RQ1.3).

Last, we want to identify what measures and factors influence the quality of OSSECOs (RQ1.4).

Table 2 Research question 1 of this thesis.

RQ1	**What is currently known about the characteristics of OSSECOs?**
RQ1.1	What are the demographic characteristics of the studies about OSSECOs?
RQ1.2	What is an OSSECO?
RQ1.3	Which representations have been proposed for OSSECOs?
RQ1.4	What measures or attributes are defined to assess or evaluate OSSECOs?

- **To design the QuESo-framework** to support the activities of the OSSECO quality evaluation, and **to validate the framework** in a real-world context. The research question to achieve this objective is formulated in RQ2 and is decomposed in several sub-questions (see Table 3). In Section 5 there is a detailed description of the QuESo-framework validation.

Table 3 Research question 2 of this thesis.

RQ2	**Is it feasible to use the QuESo-framework to evaluate the quality of OSSECOs?**
RQ2.1	What is the perceived validity of the QuESo-model? (i.e., "*The degree to which all the elements contained in the QuESo-model should actually appear in the model in the right way"* [173]).
RQ2.2	What is the validity of the results obtained applying the QuESo-process?
RQ2.3	What is the perceived usefulness of the QuESo-process? (i.e., *"The degree to which the subject considers that the evaluation process is effective in achieving its intended objectives"* [173]).

1.4 Methodological approach

To achieve the research goals and answer the research questions of this thesis, our methodological approach is based on design science methodology by Wieringa [1] and Shaw [2], who states several ways of characterizing software engineering research, in terms of research settings, research products, and validation techniques, that are described and characterized below according to the research questions of the thesis previously defined**.**

1.4.1 Research setting

The research setting classification allows to investigate and formulate the problem, i.e., the problem is transferred into a research setting with the aim of finding solutions to it. For this purpose, Shaw proposes five research setting types with some kinds of questions posed by each that are specified as follows:

- **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?

From this perspective, the research setting types that better fit the research presented in the thesis are *characterization* for RQ1 and *method/means* for RQ2, since:

→ RQ1 tries to **characterize** the OSSECOs in term of its definitions, particularities, measures and representation techniques.

→ RQ2 tries to define the **means and methods** on how to evaluate the quality of OSSECOs.

1.4.2 Research product

The research product classification represents the main outcomes of the research, i.e., allows specifying the tangible research results of the research. For this purpose, Shaw proposes five research product types described as follows:

- **Qualitative or descriptive model**. Organize & report interesting observations about the world. Create & 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 implementation 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.

From this perspective, the research products of the thesis are *qualitative or descriptive model* for RQ1, and *technique and system* for RQ2, since:

→ In RQ1 we provide **qualitative/descriptive** models by organize, analyze and report observations through systematic questions about OSSECOs.

→ In RQ2 we provide **procedures and techniques** on how to evaluate the quality of OSSECOs and also, we provide a **system** implementing these techniques.

1.4.3 Validation techniques

Finally, the validation techniques allow validate the research results to demonstrate that they satisfy the research setting previously specified. For this purpose, Shaw proposes five validation techniques and their characters of validation described as follows:

- **Persuasion.** Validation by means of a technique, design or example.
- **Implementation.** Validation by means of the implementation of a system or technique.
- **Evaluation.** Validation with respect to a descriptive model, a qualitative model or an empirical quantitative model.
- **Analysis.** Validation by means of an analytic formal model or an empirical predictive model.
- **Experience.** Validation expressed in a qualitative or descriptive model, as decision criteria or an empirical predictive model.

In this thesis, *persuasion* is used all along the thesis, using examples that illustrate the behavior of the proposed ideas or processes. For the *implementation technique*, a tool has been developed that demonstrates the approach and the framework proposed in RQ2. *Evaluations* are conducted in both RQ1 and RQ2. Finally, *experience* is used by evaluating the quality of a real-world OSSECO.

1.4.4 Research phases

To solve the design problem and answer the research questions, we followed an approach based on the design science methodology [1]. Our approach comprises

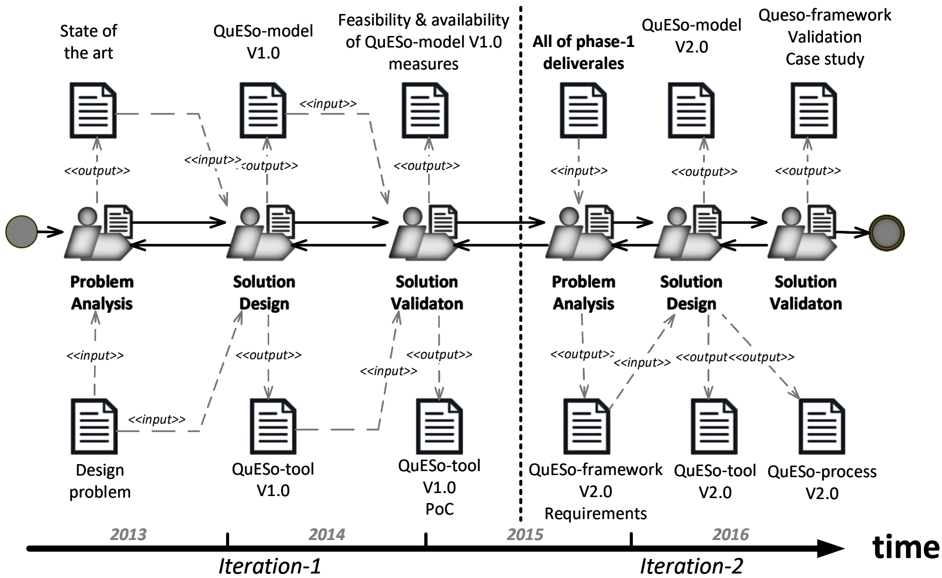

Figure 1 Research methodology phases

three phases: (i) *problem analysis*. (ii) *solution design*, and (iii) *solution validation*. These phases were conducted along two iterations, Figure 1 depicts the timeline of methodological process.

- **First iteration** (QuESo-framework V1.0): Initially, we conducted a systematic mapping on OSSECOs to analyze the problem to be solved (*phase I*). Based on the results of this study, we identified the requirements for designing the QuESo-model V1.0 and for the development of the QuESo-tool V1.0 (*phase II*). Finally, we validated these artifacts in two steps: first, we identify the suitability of the QuESo-model V1.0; second, we designed a Proof of Concept (PoC) for integrating the QuESo-tool V1.0 to the SALMon framework [53], in order to evaluate a set of OSSECO health indicators (*phase III*).
- **Second iteration** (QuESo-framework V2.0): We start with the analysis of the QuESo-model V1.0 with the purpose of identifying improvement issues. In addition, based on the analysis of the deliverables from iteration I, we derived a set of requirements for designing the QuESo-framework V2.0 (*phase I*). After this, we updated the quality model to the version QuESo-model. Additionally, we designed the QuESo-process to evaluate the quality of OSSECOs using the QuESo-framework also, we improved the QuESo-tool (*phase II*). Finally, we designed a case study to validate the QuESo-framework (*phase III*).

1.5 Contributions of this thesis

This thesis was initially motivated and supported by the FP7 European Project RISCOSS. This project was coordinated by the research group GESSI, and it was mainly focused in managing risks in OSS adoption. In this thesis, we developed several deliverables to support the RISCOSS project (see Table 4).

The main contribution of this thesis is the design of the **QuESo-framework** that supports the quality evaluation of OSSECOs. In order to contribute with this design problem, we defined two research goals, The First is an assessment of the state-of-the-art on OSSECOs. The Second is the design and validation of the QuESo-framework by showing that the framework is valid, usable and useful. The **QuESo-framework** is divided into three interrelated artefacts: (i) **QuESo-model** (the main artefact) is a quality model for measuring the quality of OSSECOs. This model defines a hierarchical structure of measures, quality subcharacteristics and quality characteristics, (ii) **QuESo-process**, is a process for the evaluation of the quality of OSSECOs using the QuESo-model and (iii) **QuESo-tool**, is a software component for semi-automatic quality evaluation of OSSECOs.

1.5.1 State of the art in OSSECOs

In order to have an unbiased view of the state-of-the-art and current research on OSSECO, we designed and conducted a systematic mapping following the guidelines described by Petersen et al. [54], [55] and Kitchenham and Charters [56]. The main goal and contribution of this systematic mapping is to provide a reference for prospective researchers and practitioners in the field specially to help avoiding the definition of new proposals that do not align with.

Furthermore, other contributions of the systematic mapping include:

- A taxonomy of the terms related to the OSSECO.
- A genealogical tree to understand the genesis of the OSSECO term from related definitions of this type of SECO.
- An analysis of the available definitions of SECO in the context of OSS.
- A classification of the existing OSSECOs modelling and analysis techniques.
- A list of measures or attributes to assess or evaluate OSSECOs.
- A research roadmap for OSSECOs

The Results of such study was published in the SCI-indexed journal Information and Software Technology (I.F.: 2.694, JCR 2016) [34]. Additionally, the list of measures was published in the conference paper [50].

1.5.2 The framework

The interrelated artefacts of the framework were designed and validated iteratively along the design process, resulting in the proposed quality evaluation framework named QuESo-framework. An earlier proposal for the framework was presented in the conference OSS-2015 [57], winner of the PhD contest in software-ecosystems.

- **QuESo-model:** The first version of the model, named **QuESo-model V1.0**, was published in the conference ICSOFT-EA 2014 (CORE-B, Best paper nomination) [50] **(Chapter 3)**. The initial attempt of the model validation was to validate the feasibility of obtaining the QuESo-model measures identified in the literature related to the GNOME ecosystem **(Section 5.1)**. An extended version of this work has been published as a book chapter in [58]. In addition, we proposed a new version of the quality model named **QuESo-model V2.0**, in order to improve several aspects such as: measure definitions, set of measures and subcharacteristics **(Chapter 3)**.
- **QuESo-process:** in order to use the QuESo-model to evaluate the quality of OSSECOs, we designed a process to aggregate OSSECO-model measures in terms of a cause-effect relationship between measures and quality subcharacteristics using Bayesian networks (BN) approach **(Chapter 4)**. This is a six-tasks process: (i) configure evaluation, (ii) gather raw data, (iii) define BN structure, (iv) assign BN prior-probabilities, (v) calculate subcharacteristics and (vi) validate results. The QuESo-process was modelled using the SPEM notation (i.e., Software & Systems Process Engineering Metamodel) [59], the SPEM model of the QuESo-process can be consulted in http://plateoss.azurewebsites.net/.
- **QuESo-tool:** we designed a software tool, in order to show the feasibility to use the QuESo-process and to support the validation of the QuESo-model. An earlier proof of concept implementation focused in monitoring a list of community health measures **(Chapter 6)** was published in the conference RCIS-2014 (CORE-B) [60] and presented in the JCIS-Sistedes 2015 [61].

The framework contributes to fill the gap of the quality operationalization of software ecosystems as follows:

- First, the **QuESo model** provided three-dimensional perspectives for evaluating OSSECOs: (1) those that relate to the platform around which the ecosystem is built; (2) those that relate to the community (or set of communities) of the ecosystem and (3) those that are related to the ecosystem as a network of interrelated elements, such as projects or companies. This is a novel approach because according to authors like Jansen [62]. Currently, there exists no working operationalization available that can be used to determine OSSECO characteristics (e.g., quality and health). Jansen argues that OSSECO quality characteristics typically looked at from a project scope, not from an ecosystem scope.

- Second, the **QuESo process** and the **QuESo tool** are based on the **QuESo model** in order to provide an operationalization of the OSSECO quality evaluation using Bayesian Networks.

The validation of the **QuESo-framework** is presented in **Chapter 5**.

1.6 List of Publications

List of publications related to the European project RISCOSS are shown in Table 5, while list of publications related to the PhD thesis are shown in Table 4

Table 4 List of publications related with the RISCOSS EU project

Reference	Venue	Title	Year	Remarks
[205]	COMPSAC WRMOCC	Expert Mining for Evaluating Risk Indicators Scenarios	2014	Workshop paper
Not referenced	INSTIC	RISCOSS: Managing Risk and Costs in Open Source Software Adoption	2015	Book chapter in edition
[206]	OSS	The RISCOSS Platform for Risk Management in Open Source Software Adoption	2015	Full paper

Table 5 List of publications related with the QuESo-framework

Reference	Venue	Title	Year	Remarks
		Book chapters		
[207]	Software Technologies	Measuring the Quality of Open Source Software Ecosystems using QuESo	2015	Published in Springer
		Conference proceedings		
[20]	ICSOFT-EA	QuESo: A Quality Model for Open Source Software Ecosystems	2014	CORE-B Best paper nomination
[60]	RCIS	Assessing Open Source Communities’ Health using Service Oriented Computing Concepts	2014	CORE-B Regular paper
[208]	OSS	Open Source Software Ecosystems: Towards a Modelling Framework	2015	Winner of the PhD Contest in SECOs
[61]	JCIS SISTEDES	Towards Assessing Open Source Communities' Health using SOC Concepts	2015	Short paper
		Technical reports		
[125]	UPC commons	QuESo a quality model for open source software ecosystems: list of measures	2015	
		Workshops		
Not referenced	iStar International Workshop	iStarJSON: A Lightweight Data-Format for i* Models	2016	Workshop paper
		Journals		
[209]	**IST**	**Open Source Software Ecosystems: A Systematic Mapping**	**2017**	**Published**
Not referenced	Not defined	Assessing the Quality of Open Source Software Ecosystems: A Case Study of Eclipse	2017	To be submitted

1.7 Structure of this thesis

The structure of the present PhD thesis is organized in chapters that have the aim to provide an abstract of the most important findings and contributions of the thesis. Each chapter is oriented to cope the research goals and research questions exposed in **Section 1.3**.

Table 6 shows the distribution of the chapter according to the Wieringa's design cycle [1]. **Chapter 2** describes a Systematic Mapping conducted to study the relevant knowledge about OSSECOs. **Section 6.1,** we implemented a Proof of Concept in order to understand the phenomena of analyzing OSSECOs through OSS community data sources. **Chapter 3**, **Chapter 4** and **Chapter 6** presents the design of the artifacts that make up the QuESo-framework. Chapter 5 provides the validation of the QuESo-framework.

Table 6 Thesis Structure

Chapter 1: Introduction	
Problem Investigation	Chapter 2: Literature study: OSSECOs
	Chapter 6 Section 1: Proof of concept: QuESo-tool
Treatment Design	Chapter 3: Quality model: QuESo-model
	Chapter 4: Framework process: QuESo-process
	Chapter 6: Support :QuESo-tool
Treatment Validation	Chapter 5: Validation: Framework Validation
Chapter 7: Conclusions and future work	

2 Literature study

"I go to seek a great perhaps"
— **Gabriel García Márquez**, **The General in His Labyrinth**

Open source software (OSS) and software ecosystems (SECOs) are two consolidated research areas in software engineering. OSS influences the way organizations develop, acquire, use and commercialize software. SECOs have emerged as a paradigm to understand dynamics and heterogeneity in collaborative software development. For this reason, SECOs appear as a valid instrument to analyze OSS systems. SECOs provide a new holistic point of view for understanding OSS. In this sense, OSS initiatives typically create an adequate environment for making a SECO emerge from their projects, communities, and external actors (partners, public and private institutions, research groups). This means that SECOs provide a practical approach to understand all of the synaptic relationships between OSS heterogeneous elements.

The aim of this first contribution of the thesis is to learn and describe relevant knowledge about OSSECOs, and to identify the most remarkable gaps to bridge (**RQ1** of his thesis). As a result of this literature study, we have been able to identify the strengths and weaknesses of the current state of the art in OSSECOs. The importance of this work summarized in this section lies mainly along two lines. First the identification of the main OSSECO definitions, OSSECO particularities and knowledge about OSSECO representation. Second, the identification of measures that influence the quality of OSSECOs.

2.1 Systematic Mapping Studies

Systematic Mapping Studies (SMs) 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 [55], [56]. They involve searching the literature in order to know what topics have been covered, and where they have been published [54], [55]. Although an SM 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, SMs 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 SM defined by Petersen et al. [54], [55] have been followed in this thesis. Whenever Petersen's et al. guidelines were not specific enough, SLR guidelines defined by Kitchenham and Charters [56] 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. [54], are as follows:

- **Planning the mapping.** Activities performed before conducting the review. 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 review, 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 refview. It includes the specification of a dissemination mechanism, the format of the report, and its evaluation.

This methodology may be applied to the two main existing types of systematic literature studies, namely systematic literature reviews (SLR) [56] and systematic mappings (SM) [54], [55]. Whilst SLRs investigate in detail some insightful research question, a systematic mapping is a method to review, classify, and structure papers related to a specific topic. The goal of a systematic mapping is to obtain an overview of existing approaches, outlining the coverage of the research field and its gaps. Given the objective of our study, it can be classified as a systematic mapping.

2.1.1 Research Method

This research is based on both the guidelines proposed by Petersen et al. [54], [55] for the development of mapping studies and those described by Kitchenham and Charters [56] for the development of systematic literature reviews. The mapping process is split into several phases (see Figure 2), which are described in the following subsections.

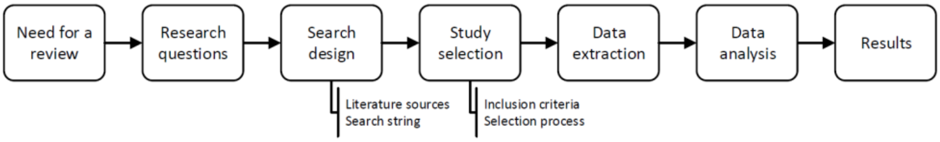

Figure 2. Stages for the systematic mapping

2.1.2 Identification of the need for a review

As Kitchenham and Petersen state, prior to undertaking any systematic literature study, researchers should identify and review any existing systematic review of the phenomenon of interest against appropriate evaluation criteria. There is no procedure defined to implement this stage. However, similar to Oriol et al.,[63], we applied two strategies. First, to broaden the scope of the results, we searched other systematic literature studies in the area of SECOs (not only opensource). Second, we followed a procedure that is analogous to the main search of our systematic mapping. In other words, we defined a search protocol to identify other secondary studies. The protocol was based on the protocol defined in the main search, which will be explained in the following sections. In short, we used the same digital libraries (see Table 9), and we built the search string as a conjunction of population and intervention as recommended by Kitchenham and Charters [56]. From each term of the population and intervention, we identified a set of variants and acronyms:

> *(“software ecosystem” OR “software ecosystems”) AND (“state of the art” OR “SLR” OR “review” OR “systematic mapping”)*

As a result of this search. we identified six secondary studies that presented a review on SECOs: Barbosa and Alves [64]; their updated work Barbosa et al. [37]; Manikas and Hansen [65]; their expanded work Manikas [39]; Axelsson and Skoglund [49] and Manikas [66]. Afterwards, we analyzed all of the selected papers from these studies and found a new study by Hanssen and Dybå [38], which is a kind of secondary study about theorizing in the SECO research literature. It is worth noting that none of these works was conducted specifically on OSSECOs. Instead, all papers focus on SECOs in general except [66]. that focuses on proprietary SECOs.

- Barbosa and Alves [64]. See Barbosa et al. [37] which expands this work.
- Hanssen and Dybå [38] described the theoretical foundations of SECOs. In their work, they identified openness and transparency as one of the fundamental concepts for further and deeper research in SECO theorizing. In addition, they presented five main theories related to SECOs: activity

theory, transaction cost theory, systems theory, sociotechnical theory, and intermediary theory.

- Barbosa et al. [37] conducted a systematic mapping study on SECOs. They defined four research questions about the characteristics, benefits, and challenges of SECOs. In their work, ten characteristics of SECOs were identified and eight main SECO research areas were found. According to Barbosa and Alves, the most relevant research areas in SECOs are open source software, ecosystem modelling, and business issues. Finally, they highlighted the relevance of OSS models in the context of SECOs.
- Manikas and Hansen [65]. See Manikas [39] which expands this work.
- Manikas [39] analyzed 231 papers collected from 2007 until 2014. He identified the term open as one of the keywords related to SECOs, which is specifically related to the domain of the studies (e.g., the OSS domain). Furthermore, he identified three signs of SECO maturity: (a) a rapid increase in the number of journal articles; (b) an increase in the empirical models within the last two years; and (c) a large set of ecosystems studied. Finally, Manikas encouraged undertaking studies of specific SECO definitions rather than wide ecosystem studies in order to address SECO complexity. Our systematic mapping is an in-depth study of aspecific type of SECO (i.e., OSSECO).
- Axelsson and Skoglund [49] investigated the challenges related to quality assurance in software ecosystems and identified what approaches have been proposed in the literature. They selected six papers covering quality assurance in software ecosystems from different perspectives. The authors, also presented a list of research challenges that are specific to quality assurance in SECOs (e.g., stakeholder requirements definition and system architectural design). In their research agenda, Axelsson and Skoglund called for more research (primarily empirical) to better understand niche player needs (such as OSS communities).
- Manikas [66] investigates literature on non-open source ecosystem studies and identifies the aspects studied in this type of SECOs.

Given the lack of secondary studies specific for the OSSECO topic and the observation that OSSECOs have specific characteristics in the context of SECOs, such as the presence of an OSS community actor, we think that conducting a systematic mapping about OSSECOs is justified. In the next subsection, we provide further details of the relationships of the research questions in these secondary studies compared to those in ours.

2.1.3 Research questions

The overall research objective of this study is to find and analyze the current state of the art in OSSECOs. This objective has been broken down into three high-level research questions (RQs) which, in turn, will drive the review method. The RQs postulated in this review are exploratory since we are attempting to understand and identify useful quality data and clarify definitions about the OSSECO phenomenon. In addition, the high-level research questions are divided into research subquestions. Table 7 shows the RQs and their motivation.

Once the RQs of this study have been formulated, we compare them with those of the secondary studies identified in Section 2.1.2 (see Table 8).

Table 7 Research questions.

Research Question	Interest and motivation	Sub-questions
SM-RQ1. What are the demographic characteristics of the studies about OSSECOs?	Identify the type of publication, in particular journals publications, and the type of papers, in particular empirical, is important because are indicators of the maturity in a new research field [39]. The evolution in the number of publications is an indication of how the activity of a research field changes [210] The information about geographical distribution of the publications and the classification between academy and industry is relevant because OSSECOs concept extend geographical and institutional boundaries. Finally, the OSSECO predominant researchers are important in order to identify the keystone authors in the growing network of OSSECO researchers	SM-RQ1.1 In which type of sources are articles mostly published? SM-RQ1.2 How has the number of publications evolved over the years? SM-RQ1.3 How are papers geographically distributed? SM-RQ1.4 Who are the predominant researchers? SM-RQ1.5 How are publications distributed between academy and industry? SM-RQ1.6 What type of papers are published?
SM-RQ2. What is an OSSECO?	OSS and SECOs are two emergent research areas in software engineering [211].Consequently, by answering this RQ, we can get information about: existing elements, definitions and general characteristics of OSS, SECO and OSSECO existing in the software engineering literature.	SM-RQ2.1 What definitions are related to the OSSECO definition? SM-RQ2.2 Are there specific definitions of OSSECO? SM-RQ2.3 What elements belong to an OSSECO? SM-RQ2.4 What instances of OSSECOs have been reported in the literature?
SM-RQ3. Which presentations have been proposed for OSSECOs?	To identify which are the representations proposed in the literature for OSSECO, identifying modelling techniques, analysis, particular notations and guidelines.	SM-RQ3.1 Which primary studies use models to represent OSSECOs? SM-RQ3.2 Which of the proposed models, if any, are specific for OSSECOs? SM-RQ3.3 Which notation and guidelines have been used for modelling OSSECOs? SM-RQ3.4 What type of analysis was conducted using the models identified in RQ3.3?

- SM-RQ2 is partially addressed by the Barbosa et al.'s RQ1: What are the main characteristics of a Software Ecosystem? [37]. However, our goal in this RQ was to find a definition for OSSECOs, which is related but different.
- SM-RQ1, SM-RQ1.2 and SM-RQ1.3 are addressed by Manikas [39]. However, this type of research questions is a usual practice in systematic reviews, according to the common guidelines for this type of study. For instance, this information can be useful as input for further studies in the field in order to establish research trends.
- Manikas' RQ: How is the term software ecosystem defined? [39] ,is similar to our research question SM-RQ2.1. However, we are searching definitions for OSSECO specifically.
- Manikas' RQ: Is software ecosystem research targeting real software ecosystems? [39], is related to our research question SM-RQ2.4. However, we are searching for instances of OSSECO specifically.

Table 8. Relationships between research questions of our study and other secondary studies.

RQ	Hansen and Dyb	Barbosa et al.	Manikas	Axelsson and Skoglund
SM-RQ1	N	N	A	N
SM-RQ1.1	N	N	A	N
SM-RQ1.2	N	N	A	N
SM-RQ1.3	N	N	N	N
SM-RQ1.4	N	N	N	N
SM-RQ1.5	N	N	N	N
SM-RQ1.6	N	N	N	N
SM-RQ2	N	PA	N	N
SM-RQ2.1	N	N	PA	N
SM-RQ2.2	N	N	N	N
SM-RQ2.3	N	N	N	N
SM-RQ2.4	N	N	PA	N
SM-RQ3	N	N	N	N
SM-RQ3.1	N	N	N	N
SM-RQ3.2	N	N	N	N
SM-RQ3.3	N	N	N	N
SM-RQ4	N	N	N	N

(N: Not addressed, PA: partially addressed, A: addressed).

In every systematic mapping, the primary studies are identified by using automatic searches on scientific bibliographies or browsing manually by gathering the works from specific known journals and conferences of the target field. In our systematic mapping, we applied an automatic search that was complemented with manual searches in the specific venues listed in Section 2.1.3.1 The aim of this search process was to find as many primary studies related to the research questions as possible using an unbiased search strategy.

2.1.3.1 Literature sources

To ensure the consideration of appropriate venues, we selected a set of publication channels. The main purpose of this selection was to double-check that automatic searches covered all of these venues. In order to do this, relevant journals and conferences were selected from previous literature reviews on software engineering, OSS, and SECOs [67]. Furthermore, we added the four systematic literature reviews about SECOs mentioned in Section 2.1.2. Finally, we decided to add the book by Jansen et al. about SECOs [68] because, based on our knowledge and that of other authors [69], this is the only book that is completely devoted to the study of the concept of SECO. We finally selected the following list of journals, conferences, and workshops:

- Journals: (Software engineering) TOSEM, ASE, Communications, Computer IEEE, IEEE Software, DKE, EMSE, Engineering & Technology[1], IEEE Review, TSE, IET[2], ISJ, IST, JSS, REJ, SPE, SoSyM, SPIP[3]. (OSSECOs and Information systems and management) First Monday, Information Technology & People, IJOSSP, Journal of Industrial Economics, Knowledge Technology and Policy, Long Range Planning, Management Science, MIS Quarterly Executive, Research Policy.
- Conferences and workshops: (Software engineering) ASE, CAISE, COMPSAC, ESEC/FSE, ESEM, HICSS, ICSE, ISESE, METRICS, RE, SAC, SEKE. (OSSECOs and Information systems and management) FOSDEM, IWSECO, OS- CON, OSS, IFIP, WOSSE-ICSE, WFLOSS-ICSE, WSKS.

Note that these sources represent the main corpus whose exploration needs to be enforced by the systematic mapping. However, since we will use digital libraries (see Section 2.1.4), papers published in other venues will eventually be found.

[1] Previously IEEE Software Proceedings

[2] Previously IEE Review

[3] From 2010 incorporated in Journal of Software Maintenance and Evolution (last issue December 2011)

2.1.3.2 Search string

The aim of our search string is to capture all of the results that relate to the research questions. According to Kitchenham and Charters [56], a good way to create the search string is to structure it in terms of population, intervention, comparison, and outcome. However, similar to [70], we focused on the population dimension. Since, in fact, we are interested in two areas simultaneously, the search string is a conjunction of the two corresponding populations:

> *search string = OSS population AND Ecosystem population*

There are several terms for OSS (see Section 1.1.1). The potential OSS synonyms have been identified from [71]: "Free Software", "Libre Software", and the commonly used acronyms OSS, FOSS, and FLOSS. All of these are included in the search string to capture all relevant literature. In contrast, for the word ecosystem,

> *(OSS OR FOSS OR FLOSS OR "Open Source" OR Free Software OR "Libre Software") AND (ecosystem OR "Software Supply Network" OR "Software Supply Industry")*

we have identified "Software Supply Network" from [72]. He uses this term to define a network of linked, software products, hardware, and services to satisfy market demands. In addition, we have used the term "Software Supply Industry" from Messerschmitt and Szypersky's book [73]. The resulting query string was:

2.1.4 Study selection

The study selection strategy was designed to consist of a set of several steps, which is an adaptation of the steps proposed in [55] and [56]. Figure 3 presents an overview of the study selection process and the number of publications included in each stage. The details of each stage are described in the following subsections. We excluded articles based on titles and abstracts as well as full-text reading.

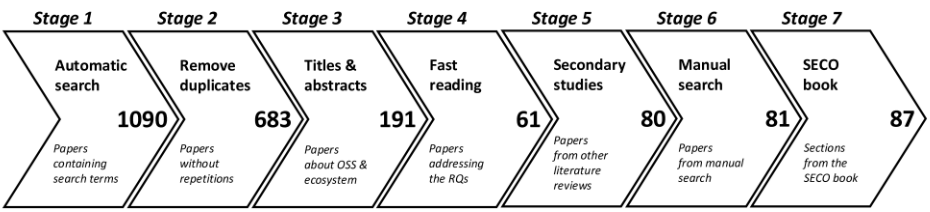

Figure 3 Stages of the selection process.

2.1.4.1 Inclusion and exclusion criteria

The following criteria have been used to select the relevant publications:

- only publications in English.
- only papers published between 2003 (publication of the seminal book about SECOs) and 2015.
- only papers about OSSECO topics. We excluded panels, prefaces of conferences and special issues, book reviews, news flashes, short papers (fewer than 4 pages), and PhD symposium papers, (i.e., publications without bibliographic information, papers that only report work in progress, and non-peer reviewed publications).

2.1.4.2 Stage 1 - Automatic search

In this stage, we identified a set of publications that serve as a basis for this study. For the selection of digital libraries, we determined a set of representative digital libraries (see Table 9) that cover the publication sources in Section 2.1.3.1. We executed the search on each digital library[4] and saved the references in bibliography files. As a result, 1090 primary studies were identified.

Table 9. Digital libraries.

Library	Link
ACM Digital library	dl.acm.org
Compendex/Inspec	www.engineeringvillage2.org
IEEE Xplore Digital Library	www.ieeexplore.ieee.org
Sciencedirect	www.sciencedirect.com
SpringerLink	www.springerlink.com
Willey Online Library	onlinelibrary.wiley.com

2.1.4.3 Stage 2 - Remove duplicates

Duplicate records were resolved in this stage by importing the references to a reference manager system and automatically removing duplicated papers. Finally, one of the authors manually reviewed the list of articles in order to identify duplicated records. A total of 407 papers were excluded in this stage (e.g., Figay and Ghodous [11] is indexed in both IEEE and ACM digital libraries).

[4] the last automatic search was made on February 12th, 2017

2.1.4.4 Stage 3 - Titles and abstracts

To identify publications that were indeed about OSSECOs, all of the authors of this study reviewed all of the titles and abstracts and checked the inclusion and exclusion criteria for each entry. When there was a disagreement, the authors discussed the issues until a consensus was reached. After this stage, 492 out of the 683 remaining papers were discarded, resulting in 191 publications (e.g., [260] was discarded because the title and abstract were not related to OSSECOs).

2.1.4.5 Stage 4 - Fast reading

Then, in order to filter out the papers from the third stage, the results and conclusions of each study were reviewed, and each researcher briefly studied their contents. Hence, all of the papers that did not reflect the study's topics, did not address any of the research questions, or were delta papers 8 were excluded (e.g., [112] is delta paper of [R2]). At the end of this stage, 61 papers were selected.

2.1.4.6 Stage 5 - Secondary studies

Thereafter, in order to identify the maximum number of relevant papers that might have been missed, we reviewed the papers from the seven secondary studies (see Section 2.1.2). In this stage, we included 19 papers out of the 315 papers referenced by the secondary studies. These papers underwent the same process that we used for the rest of the papers from Stage 2 to Stage 4.

2.1.4.7 Stage 6 - Manual search

We complemented the search in the digital libraries with some manual searches in order to ensure that we had covered all of the editions of the literature sources listed in Section 2.1.3.1. One paper was identified using this manual search process (i.e., Morgan et al. [74]).

2.1.4.8 Stage 7 - The SECO book

Jansen et al. [68] published their book: Software Ecosystems Analyzing and Managing Business Networks in the Software Industry. We applied Stage 2 to Stage 4 to the book chapters and selected six additional relevant studies.

2.1.4.9 Final result

Finally, after this last stage, the systematic mapping included 87 relevant papers (see Table 10). These papers were coded with the prefix *R*.

2.1.4.10 Data extraction

We mainly used a qualitative data analysis approach based on the method of Miles et. al., [75] to extract the data from the selected studies. This process was conducted with the support of a qualitative data analysis tool called Atlas.ti®[5] to ensure consistent and accurate extraction of the key information related to the research questions. The extraction was performed by one of authors and reviewed and confirmed by the other three authors. We also frequently used consensus meetings to review the extracted data. Having other authors check the extraction is a common practice in systematic reviews for social sciences (Petersen et al., 2015; Hauge et al., 2010). The stages of the qualitative data analysis process were the following:

- **Data processing and preparation**: The 87 studies included in our systematic mapping were grouped into one Atlas.ti® hermeneutic unit[6].
- **First cycle, (codes and coding):** Codes are labels that assign symbolic meaning to the descriptive or inferential information. They are primarily, but not exclusively, used to retrieve and categorize similar data chunks so that the researcher can quickly find, pull out, and cluster the segments relating to a specific research question [75]. We defined a list of codes from the research questions (i.e., deductive coding). In Section 2.1.5, we detail the information that we used to define the initial codes.
- **Second cycle, (pattern codes):** This is a way of grouping the list of codes into a smaller number of categories (i.e., pattern codes). These are explanatory or inferential codes that identify an emergent theme, configuration, or explanation [75]. In Section 2.2, we describe these categories in the research questions where they were defined.
- **Displaying the data:** The goal of this stage is to condense the major data and findings from our study for further analysis and to represent and present the conclusions. In our study, we used different kind of methods to display the results (e.g., tables and charts).

[5] http://atlasti.com

[6] hermeneutic unit is an Atlas.ti container where all of the information, links, or paths to this information that are related to a specific project, are stored.

Table 10 Overview of selected studies.

ID	T	Year	Country	C	Scope	ID	T	Year	Country	C	Scope
R1	J	2008	Sweden	R	BECO	**R45**	J	2010	USA	N	
R2	C	2011	Brazil	E	DBECO	**R46**	W	2009	USA	N	BECO
R3	C	2010	USA	N		**R47**	J	2013	Ireland	N	BECO
R4	C	2011	Turkey	E	BECO	**R48**	B	2013	Netherlands	R	BECO
R5	C	2006	UK	R		**R49**	B	2013	Finland	N	BECO
R6	B	2006	USA	R	SECO	**R50**	B	2013	Netherlands	R	
R7	C	2009	France	E		**R51**	B	2013	Netherlands	R	
R8	C	2010	Sweden	E	SECO	**R52**	B	2013	Netherlands	N	BECO
R9	C	2011	Sweden	R		**R53**	B	2013	Belgium	R	SECO
R10	C	2010	Belgium	N	SECO	**R54**	C	2014	Belgium	E	
R11	C	2011	USA	R		**R55**	C	2013	Belgium	E	
R12	C	2011	Netherlands	R	SECO	**R56**	W	2013	Denmark	N	SECO
R13	J	2012	Finland	R	BECO/ SECO	**R57**	C	2013	Japan	E	SECO
R14	J	2010	Switzerland	N	SECO	**R58**	C	2013	Luxembourg	E	
R15	C	2012	USA	E	BECO	**R59**	C	2014	Finland		BECO
R16	C	2011	Belgium	N	BECO	**R60**	C	2013	Netherlands	E	
R17	C	2011	Japan	R	BECO	**R61**	J	2013	Canada	R	SECO
R18	C	2009	Sweden	N		**R62**	J	2014	Netherlands	R	BECO
R19	C	2012	Belgium	E	SECO	**R63**	W	2013	Finland	R	
R20	J	2007	Germany	N		**R64**	J	2014	Netherlands	R	
R21	J	2006	USA	N		**R65**	C	2013	Netherlands	R	SECO
R22	J	2009	Spain	N	SECO	**R67**	C	2013	Canada	R	
R23	C	2007	USA	N		**R68**	J	2013	USA	N	BECO
R24	J	2012	USA	E	SECO	**R69**	W	2014	Netherlands	E	BECO
R25	J	2012	UK	E	SECO	**R70**	J	2014	Canada	R	SECO
R26	C	2012	USA	N		**R71**	C	2014	Sweden	E	BECO
R27	J	2010	UK	E	DBECO	**R72**	W	2015	Spain	E	BECO
R28	C	2010	Germany	R	SECO	**R73**	W	2014	Netherlands	R	
R29	C	2007	UK	N		**R74**	J	2014	Sweden	R	SECO
R30	C	2011	Netherlands	R		**R75**	C	2014	Sweden	R	BECO
R31	C	2011	Canada	E	SECO	**R76**	B	2014	Belgium	R	ECO
R32	C	2008	USA	R	BECO/ SECO	**R77**	J	2015	Finland	R	BECO
R33	C	2011	Japan	N	BECO	**R78**	C	2013	Netherlands	R	
R34	J	2012	USA	E	BECO/ SECO	**R79**	C	2009	Sweden	N	BECO
R35	B	2007	USA	N	SECO	**R80**	J	2004	USA	N	
R36	C	2011	Netherlands	R	BECO	**R81**	B	2003	UK	N	
R37	B	2013	Belgium	R	SECO	**R82**	J	1993	USA	N	
R38	J	2008	USA	N	BECO	**R83**	W	2011	Canada	E	BECO
R39	C	2009	Switzerland	E	SECO	**R84**	C	2014	Spain	E	BECO
R40	C	2011	Germany	N		**R85**	C	2014	Canada	E	SECO
R41	J	2012	Netherlands	R	BECO	**R86**	C	2014	Sweden	E	SECO
R42	C	2011	Switzerland	E		**R87**	C	2014	Finland	E	SECO
R43	J	2011	Germany	R		**R88**	C	2015	Brazil	E	SECO
R44	C	2010	Ireland	E	SECO						

To answer SM-RQ1.4, we used social network analysis (SNA) because it is useful to assess authors' positions in the social networks (this is detailed in Section 2.2.1). Finally, the process was developed based on several rounds of piloting and coding to ensure the validity and consistency of the results. To extract data from the identified primary studies, we developed the template shown in Table 11, which provides the initial codes for the data extraction process.

Table 11. Data extracted from each study.

Data item
Source (Conference, Journal, Book chapter) and full reference
Year when the paper was published
Author(s) and their affiliation (organization and country)
Type of publication
Definition(s), sources and authors of ecosystem term(s)
Elements related to OSSECO: name and type (defined, referenced, used)
Measures, if any, defined to evaluate OSSECOs
Instances, if any, of OSSECOs studied
Ecosystem model(s), if any, used
Scope of the ecosystem model(s) (SECO, BECO, DBECO)
Techniques and notations for modelling OSSECOs
Type of OSSECO analysis

2.1.5 Data Analysis

The information for each item extracted was tabulated and visually illustrated (see Section 2.2). Table 12 shows the data that was tabulated to answer the research questions.

Table 12. Data Tabulated per Research Question.

Data item	RQ
Number of papers per source	RQ1.1
Number of relevant publications per year	RQ1.2
Number of papers per country	RQ1.3
Social network measures (e.g., betweenness centrality)	RQ1.4
Number of papers of academy and industry	RQ1.5
Number of papers per type (e.g., experience report)	RQ1.6
Number of papers per type of ecosystem definition sources	RQ2
The sources of ecosystem definitions	RQ2.1
The ecosystem concept definitions	RQ2.1
The definitions of OSSECOs	RQ2.2
Number of papers per ecosystem terms	RQ2.3
Number of papers per OSSECO instances	RQ2.4
Number of papers using models to represent OSSECOs	RQ3.1
The type of SECO modelled	RQ3.2
Number of papers per modelling technique	RQ3.3
Identify the type of OSSECO analysis	RQ3.4

2.2 Results

This section summarizes the results obtained from the data extraction process.

2.2.1 SM-RQ1. What are the demographic characteristics of the studies about OSSECOs?

To answer this question, we applied the process defined in Section 2.1.4.10 without the pattern code cycle.

2.2.1.1 SM-RQ1.1 In which type of sources are articles mostly published?

The distribution of the 87 primary studies is shown in Figure 5. According to our data, conference proceedings (with 45 papers) are the most prevalent publication type. Table 10 Overview of selected studies. shows the publication type for each paper.

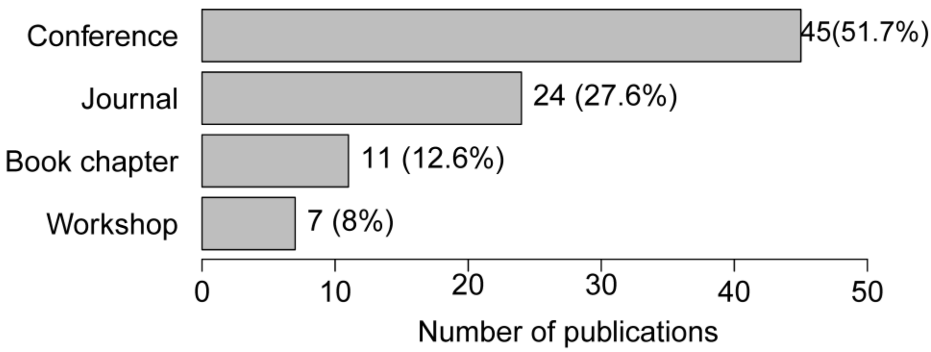

Figure 4 Publication type.

2.2.1.2 SM-RQ1.2 How has the number of publications evolved over the years?

We searched for primary studies between the years 2003-2015. We found the first papers published about OSSECOs in 2006 [R5, R6, R21]. Fig. 4 shows the number of papers per year.

2.2.1.3 SM-RQ1.3 How are papers geographically distributed?

We determined the geographical distribution of the papers based on the country of affiliation of the first author. Europe (59 papers) is the most dominant continent with the Benelux countries (24 papers) and Scandinavia (9 papers) being the most active regions (see 2.3.1 for details). North America is next with 23 papers.

There are few publications from Asia (3 papers, from Japan). Publications from other continents are scarce.

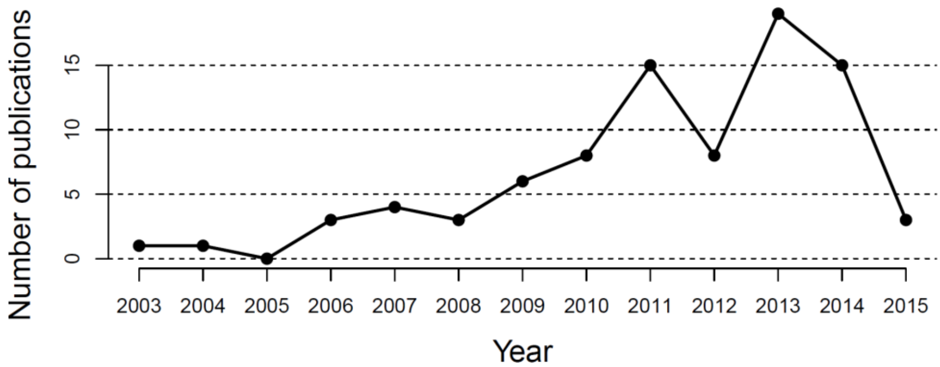

Figure 5. Publication year.

2.2.1.4 SM-RQ1.4 Who are the predominant researchers?

We addressed this question by conducting a social network analysis (SNA), which allowed us to do the following: 1) to identify individual nodes that are of particular interest (i.e., relevant authors); and 2) analyze the whole graph and identify cohesive subgroups (i.e., authors' clusters)[7]. This analysis was done only for the authors and coauthors of papers in our set of primary studies.

Identify the predominant researchers. We used an approach similar to [76] to evaluate the authors' relevance SNA. In that work, they proved that centrality measures are the best ones to assess the social significance of a cluster of authors. According to their work, the social model is represented by a non-directed graph $G = (V, E)$, where V nodes correspond to authors. The set of edges $E \subseteq E \times E$ represents the social relationships connecting authors. First, we identified 151 researchers from the papers. Second, we identified the set E of edges as follows: 1) $(a_i, a_j) \in E$ if $(a_i, a_j) \in V$ and (a_i, a_j) have coauthored a paper; 2) $(a_i, a_j) \in E$ if $(a_i, a_j) \in V$ and author a_j is cited by a_i. Finally, we calculated the following measures to rank the authors:

- The *betweenness centrality* for a node N is the sum of the fractions of shortest paths that include N for every pair of nodes in the network. If a high

[7] We used a tool named NodeXL [196]to perform the network analysis.

betweenness node is removed, a number of links may get disconnected [77]. This measure quantifies the ability of a node to act as a mediator in the network [78].

- Mathematically, *eigenvector centrality* is the first eigenvector of the adjacency matrix. The main principle is that links from important nodes are worth more than links from unimportant nodes [79]. High eigenvector centrality nodes can be leaders of the networks [77]. This measure scores nodes based on the principle that relationships with more important nodes confer more importance than relationships with less important nodes [28].
- *PageRank* measures the importance of a node within the network using a link analysis algorithm. It can be calculated using a simple iterative algorithm and corresponds to the principal eigenvector of the normalized link matrix of the network [80]. This measure score distinguishes the authority of each author in the social network [76].

Table Table 13. Top authors ranked by social relevance. lists the top 10 authors ranked using these measures.

Table 13. Top authors ranked by social relevance.

Betweenness centrality		Eigenvector centrality		Page rank	
Slinger Jansen	3519,859	Slinger Jansen	0,039	Slinger Jansen	6,087
James Herbsleb	1938,617	Sjaak Brinkkemper	0,030	Sjaak Brinkkemper	4,094
Tom Mens	1051,622	Tom Mens	0,026	James Herbsleb	3,905
Sjaak Brinkkemper	958,226	James Herbsleb	0,025	Tom Mens	3,119
Donald Wynn, Jr	892,983	Mathieu Goeminne	0,025	Daniel M. German	2,826
Daniel M. German	823,393	Daniel M. German	0,023	Mathieu Goeminne	2,682
Mathieu Goeminne	783,677	Walt Scacchi	0,022	Walt Scacchi	2,420
Lopez-Fernandez	584,000	K. Manikas	0,020	Brian Fitzgerald	2,157
Walt Scacchi	545,873	K. Hansen	0,020	Donald Wynn	2,150
Brian Fitzgerald	467,957	L. Luinenburg	0,018	Mircea Lungu	2,123

Cluster of authors. As in the previous case, we used the social model represented by a graph $G = (V, E)$, where nodes correspond to authors. However, in the current case, the set of edges E only connects two authors when they are paper coauthors in at least one publication. Then, we used the Newman algorithm [81] implemented in NodeXL to identify authors clusters. (i.e., a set of at least two authors who collaborated on at least one publication). Table 14 summarizes the authors and coauthors network-wide measures.

Table 14. Authors and coauthors overall graph measures.

Measure	Value
Number of Vertexes	180
Number of Edges	256
Number of Clusters	42
Maximum Edges in a Connected Component	40
Maximum Vertexes in a Connected Component	24
Graph Density	0.016

Figure 6 shows the three most populated clusters identified. The graphs highlight the top authors in each cluster (according to the SNA measures) and the number of relationships between coauthors.

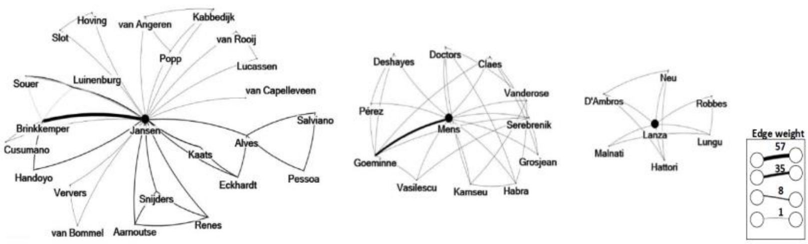

Figure 6. Research clusters (edge weight is the number of relationships).

2.2.1.5 SM-RQ1.5 How are publications distributed between academy and industry?

In order to answer this question, we analyzed whether at least one of the authors in each paper came from a non-academic institution (similarly to [70]). A total of 25 papers (28.7%) fall into this category, while 62 papers (71.3%) have authors solely from academy. We found that two papers [R18, R20] are exclusively from industry.

2.2.1.6 SM-RQ1.6 What type of papers are published?

To answer this question, we classified the publications into three categories, similarly to [71] and [82]: (R) empirical research papers, where the authors present evidence from a research study having an explicit research question; (E) experience reports, where the authors report experiences without having defined an explicit research question; and (N) non-empirical papers, which include opinion papers and theoretical papers. Figure 7 presents the number of papers of each type. The classification for each type of study is shown in Table 10.

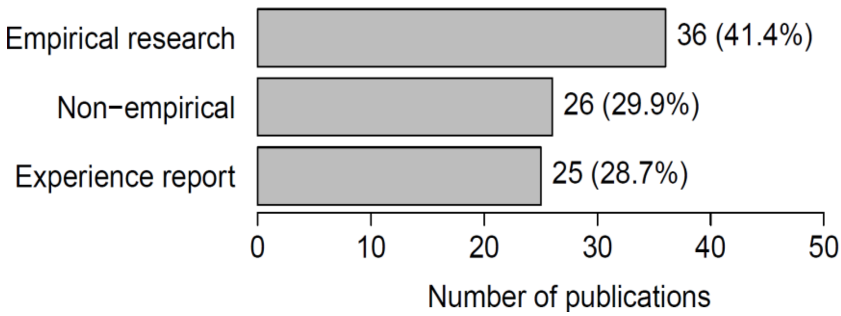

Figure 7. Type of research.

2.2.2 SM-RQ2. What is an OSSECO?

To answer this question, we applied the process defined in Section 2.1.4.10.

2.2.2.1 SM-RQ2.1 What definitions are related to the OSSECO definition?

We found that 76 papers out of the total of 87 use ecosystem-related definitions based on the five different concepts introduced in Section 2.2.2.1 shown at the top of the Figure 8. Figure 8-a shows the percentage of references of each ecosystem definition (calculated on the 76 papers that used ecosystem definitions). Figure 8-b shows the list of the papers classified by ecosystem definition.

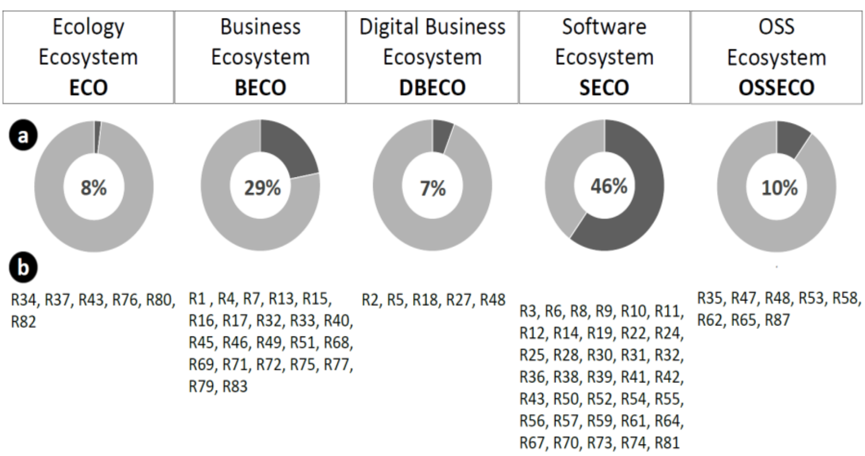

Figure 8. OSSECO classification.

Figure 8 shows that the SECO definition is the one that is most studied in the selected papers (36 out of the 87studies), the BECO definition is the second most studied (22 studies), the OSSECO definition has eight papers mentioning it, and ecosystem and DBECO[8] are the two least mentioned definitions (6 and 5 papers, respectively), Finally, 11 studies did not fit any of the classifications (i.e., [R20, R21, R23, R26, R29, R44, R63, R84, R85, R86, R88]).

2.2.2.2 SM-RQ2.2 Are there specific definitions of OSSECO?

We identified that there have only been a few attempts in the literature to specifically define what OSSECO is. Specifically, in this mapping study, we obtained only eight papers out of the total of 87 papers that use the OSSECO definition (see Figure 8). Only three of them give a definition of OSSECOs. On the other hand, 49 papers based their work on definitions related to the ecosystem related definitions (i.e., BECO, DBECO, SECO); 17 of these papers also provide definitions related to ecosystems in their own words. The three definitions of OSSECO correspond to the papers [R35. R48, R65] (see Table 15).

Table 15. OSSECO definitions.

Definition
"An arrangement of individual and organizational units, involved in or affecting the circulation, transformation, and accumulation of capital (in various forms) in order to provide cooperative development, testing, marketing, distribution, implementation, and support of open source software." [31] *[R35]*
"An OSS ecosystem is one where it is possible to add contributions to a project, create and publish components in the extension market, etc., without any barriers. Jansen et al. (2013) [R48]
"A set of developers functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. The result of the interaction is freely available for everyone." Hoving et al. (2013) [R65]

2.2.2.3 SM-RQ2.3 What elements belong to an OSSECO?

To answer this question, we applied the process defined in Section 26. First, we collected 64 related terms in the coding cycle belonging to OSSECOs[9]. Among which project, community, and developer are the top three most used terms. Second,

[8] The papers [R2, R48] are classified in this definition; however, they use the concept of digital ecosystem instead of DBECO.

[9] The list of all the terms and definitions that we found in the primary studies are in http://www.essi.upc.edu/˜gessi/PLATEOSS/.

in the pattern codes cycle, we classified the OSSECO terms into three categories, based on the type of term used in the study (i.e., the term was defined by the author), the article references to another author (in order to include the term definition), and the term is used in the article but is not defined. Table 16 shows the terms and the number of papers per category.

2.2.2.4 SM-RQ2.4 What instances of OSSECOs have been reported in the literature?

To answer this question, we identified the specific OSS communities studied in each paper. We found that 49 papers out of 87 studied specific OSS communities. Most of them studied the Eclipse ecosystem (16 papers) and the GNOME ecosystem (10 papers). The rest of the OSS-communities were studied by only one or two papers (except Ubuntu and Ruby, with 3 papers each one

2.2.3 SM-RQ3. Which representations have been proposed?

To answer this question, we applied only the first cycle of the process explained in Section 2.1.4.10. We used the codes defined for SM-RQ3 in the data extraction section (see Table 11).

2.2.3.1 SM-RQ3.1 Which primary studies use models to represent OSSECOs?

Figure 9 shows that 56 papers (64.4%) of the 87 studies in the collected literature use models to represent the actors, resources, and their relationships in the specific OSSECO under study. Table 17 shows the name and the sources of all the OSSECO instances studied.

Table 16. OSSECO terms.

Term	Own	Other	Use	Term	Own	Other	Use
Project	9	2	66	Survey	0	0	18
Community	27	5	42	Author	0	5	10
Developer	8	5	51	Keystone player	7	6	2
Platform	8	2	49	Node	10	1	5
Source Code	0	0	59	Integrator	4	1	9
Contributor	18	1	34	Adopter	5	0	7
Product	6	1	48	Artefact	1	1	10
Service	6	0	47	Niche Player	4	4	3
Repository	9	0	37	Practitioner	3	1	7
Feature	4	3	39	Behavior	0	0	9
Market	3	4	37	Reseller	1	0	8
Bug	0	0	43	Email	0	0	8
Reviewer	3	5	32	Platform Provider	3	1	4
Roadmap	2	0	37	Active User	3	0	4
License	10	1	25	Transactions	1	0	6
Partner	5	0	31	Coordinator	2	0	5
Mailing List	4	0	33	IRC	0	0	6
Foundation	12	6	16	Bug Fixer	0	4	1
Measure	1	5	28	Passive User	1	0	4
Dependency	7	0	27	Dominator	3	0	1
Member	6	2	26	Sub Community	1	0	4
Actor	10	3	20	Vocabulary	1	0	3
Stakeholder	4	1	25	Community Manager	0	0	3
Bug Tracking	1	0	29	Forge	2	0	1
Commit	3	1	24	Bug Reporter	0	2	1
Event	4	0	23	Entropy	1	0	2
Goal	3	0	21	Translator	1	1	1
Niche	15	2	6	Committer	0	0	1
Boundary	2	2	17	Configurator	0	0	1
Social Network	7	2	13	Supplier	0	0	1
Edge	16	0	5	Super Repository	1	0	0
Data Source	4	0	16	Wishlist	1	0	0

•

Table 17. SECOs Instances and Papers Related.

OSSECO	Papers
Eclipse	R9, R17, R28, R31, R34, R37, R40, R41, R43, R44, R48, R67, R68, R69, R73, R36
GNOME	R10, R11, R19, R39, R42, R53, R60, R62, R64, R76
Android	R48, R70, R86
Ubuntu	R16, R48, R50
Ruby	R12, R48, R87
Open Design Alliance ODA	R36, R41
Debian	R16, R30
Python	R62, R65
Wordpress	R48, R51
Brazilian Public Software (BPS)	R2, R8
Apache	R85
Ecos	R70
Evergreen	R35
FOSS4G	R22
Gurux Software	R13
Moodbile	R84
Nagios	R8
NASA Earth science	R15
OSAMI Consortium	R4
OSGeo	R25
OSMOSOFT	R47
OpenStack	R77
R	R67
Topcased	R9
Vaadin	R13
Webkit	R58

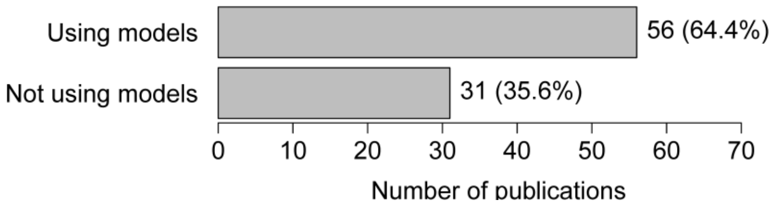

Figure 9. Use of models.

2.2.3.2 SM-RQ3.2 Which of the proposed models, if any, are specific for OSSECOs?

According to our review, none of the 56 studies using models develops a specific technique for modelling OSSECOs although most of them studied OSS communities. Figure 10 shows the different definitions of these studies. The SECO and BECO definitions are the most frequently used ones. In contrast, the DBECO definition is used in two papers, and the ecosystem definition is used in only one paper. Table 18 shows the type of ecosystem definition of each paper.

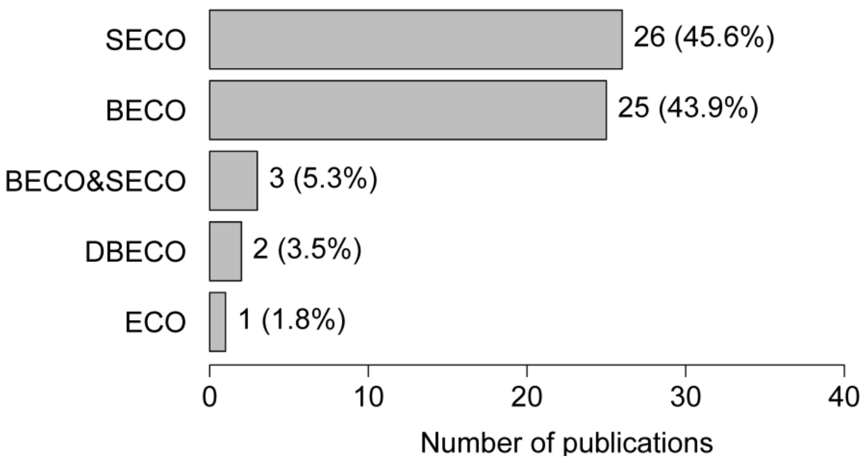

Figure 10. Use of models by ecosystem definition.

2.2.3.3 SM-RQ3.3 Which notation and guidelines have been used for modelling OSSECOs?

To answer this question, we applied the second cycle of the process explained in Section 2.1.4.10 to the results of the SM-RQ3.1. In the 56 papers that use models, we found several modelling techniques to describe or visualize software ecosystems: conceptual maps (e.g., R2, R15, R47, R69), tabular representations (e.g., R35, R52, R62, R79), mathematical notations (e.g., R6, R31, R34), metamodels (e.g., R12, R16, R39), social networks (e.g., R8, R25, R59, R77, R86, R87), class diagrams (e.g., R36, R46, R65), iStar (e.g., R72, R83), and also ad hoc notations (e.g., R22, R27, R57, R75, R88). When a paper used more than one type of modelling technique, we selected the dominant one. Figure 12depicts examples of OSSECO models according to each type of modelling technique. Figure 11 shows the distribution of papers by modelling technique. It shows that ad hoc notations (31 papers) are predominantly used to model OSSECOs. Table 18 lists the modelling technique used for each paper and the goal pursued by the model.

Table 18. Model techniques and goals.

ID	Model goal	Technique
R1	To use open source development model as a global sourcing strategy.	Tabular
R2	To characterize brazilian public software ecosystem.	Conceptual map
R4	To defining the foundations of a crossplatform open-services ecosystem.	Ad hoc
R6	To develop a model to compare the incentives to invest in operating system under open	Mathematical
R8	To elaborate approaches for how involvement of different roles can be analysed through	SNA
R10	To automate the analysis of the evolution of software ecosystems	Tabular
R12	To presents an overview of the open source Ruby ecosystem	Metamodel
R13	To propose the OSCOMM framework for studying the problem of building open source	Ad hoc
R14	To show how developers collaborate with each other within an software ecosystem across	Metamodel
R15	To model the NASA Earth science data systems ecosystem.	Conceptual
R16	To support modeling and evolution of quality from different points of view.	Metamodel
R17	To illustrate a co-creation process model of the Eclipse software ecosystem.	Ad hoc
R19	To study the GNOME ecosystem and developer community	Ad hoc
R22	To model the relationships between FOSS4G software ecosystem projects.	Ad hoc
R24	To model the software ecosystem that arise for open architecture systems.	Class diagram
R25	To map out the social history of collaborative activities within the OSGeo ecosystem	SNA
R27	To show the interactions in digital business ecosystems (as part of DBE European project).	Ad hoc
R28	To show the practices users have developed to manage the antagonism of maintaining a	Ad hoc
R31	To identify a model linking factors affecting the economics of collectives, and develop to	Mathematical
R32	To model the resilience of an organizational OSS ecosystem.	Ad hoc
R33	To propose a three-layer view model of a software ecosystem.	Ad hoc
R34	To model Eclipse platform project ecosystem.	Mathematical
R35	To propose a framework for assessing the three dimensions of software ecosystem health.	Tabular
R36	To present a conceptual overview that describes the structure of an ecosystem associated	Class diagram
R37	To Analyse the evolution of social aspects of open source software ecosystems.	SNA
R38	To show some possible symbiotic relations between Linux and other software systems.	Ad hoc
R39	To present the software ecosystem metamodel that the small project observatory	Metamodel
R41	To present the open software enterprise model that enable to establish the degree of	Tabular
R44	To propose a framework for sustainable software ecosystem management.	Tabular
R46	To propose a structure for modelling ecosystem software licenses.	Class diagram
R47	To construct a model to theorize how firms create and capture value from OSS.	Conceptual
R48	To propose a model for classifying software ecosystems.	Tabular
R49	To illustrate the management practices in technology and innovation management	Ad hoc
R52	To contribute to the concept of BECO health.	Tabular
R53	To propose a framework that enable the empirical study of OSS ecosystem and their	Tabular
R56	To propose a software ecosystem health framework.	Ad hoc
R57	To present a model for creating and sustaining communities on the information services	Ad hoc
R59	To observe how key events in the mobile device industry have affected the WebKit	SNA
R61	To present a conceptual model of the collaboration management process in a OSS	Conceptual
R62	To propose a framework for the OSS ecosystem health operationalization.	Tabular
R65	To present an analysis of the Python egg software ecosystem.	Class diagram
R68	To analise the market-driven view of an OSS ecosystem.	Ad hoc
R69	To analyse the partnership model of the Eclipse ecosystem and the activity of different	Conceptual
R70	To address an exploratory study of the solutions to variability in software ecosystem.	Ad hoc
R71	To present ESAO model, It is focused on analysing and alignment between all the different	Ad hoc
R72	To use i* roles in OSS adoption strategy models.	i*
R74	To present an open software ecosystem for embedded devices.	Ad hoc
R75	To present a study about Key Performance indicators (KPI) for software-based	Statistical
R76	To analyse the differences and analogies between natural ecosystems and software	Statistical
R77	To explore the role of groups, sub-communities and business models within a high-	SNA
R79	To present a software ecosystem taxonomy.	Tabular
R83	To ilustrate how strategic modelling using the i* framework can help in analysing different	i*
R84	To develop methodologies for managing risks of FLOSS adoption and deployment in	i*
R86	To analyze committers' networks.	SNA
R87	To verify whether the SECO context maintains the high socio-technical congruence levels	SNA
R88	To propose a collective intelligence (CI) approach for improving the free software adoption	Ad hoc

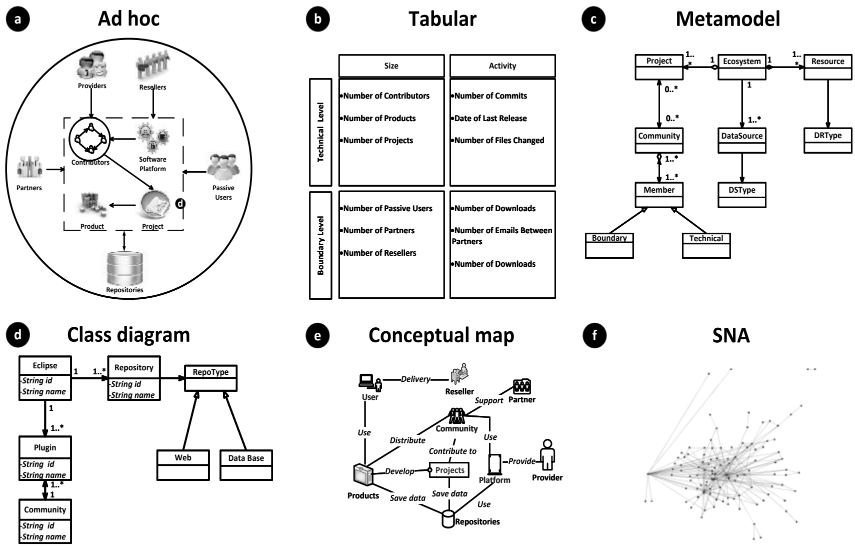

Figure 12. Examples of modelling techniques: OSSECO models a, c, d, and b are static; model f is an OSSECO dynamic model; and figure b shows a level-oriented framework model

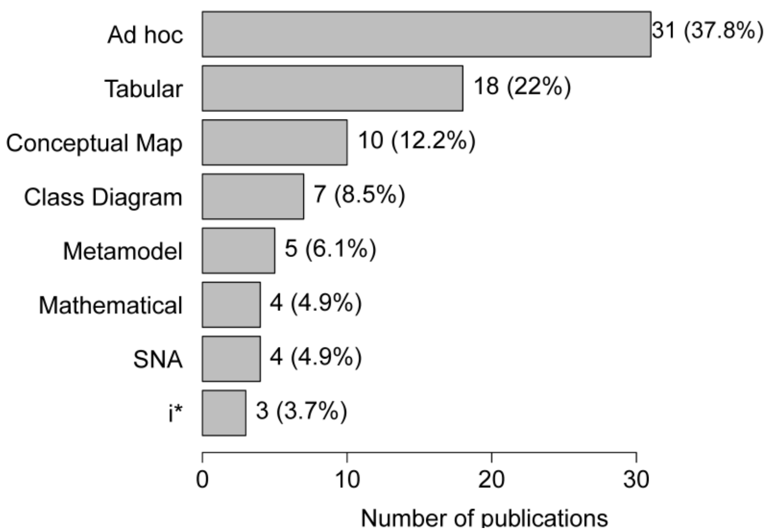

Figure 11. Modelling techniques and notations.

2.2.3.4 SM-RQ3.4 What type of analysis was conducted using the models identified in SM-RQ3.3?

We classified the papers that use OSSECO models into four categories (i.e., social network analysis, statistical analysis, visual analysis, and mathematical analysis). This classification is based on the approach proposed by [83]. Figure 13 shows the distribution of selected studies that use models in the type of OSSECO analysis. Twenty-three of these studies (41.1%) do not show any evidence of analysis.

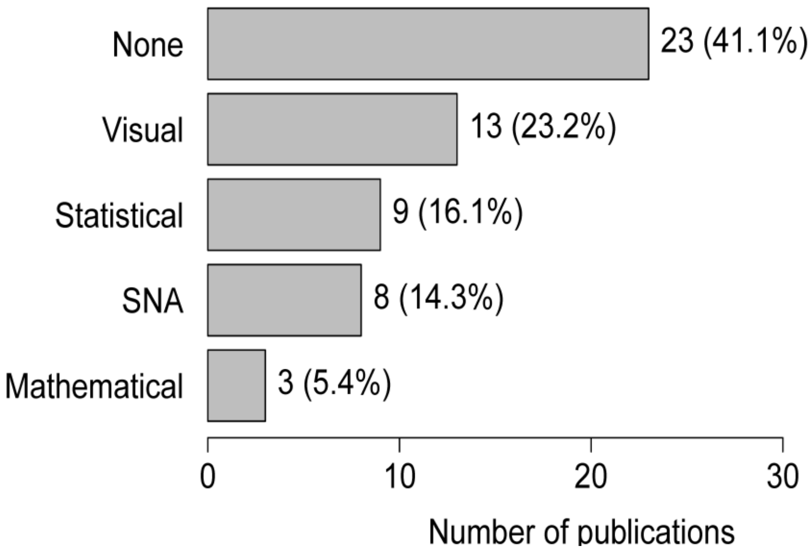

Figure 13. Analysis techniques.

2.3 Discussion

In this section, we discuss each of the answers to our research questions. For the analysis of some of the results, we performed a correlation analysis between all the codes used to answer the research questions. However, in this study we only considered the statistically significant correlations[10]. The independence test used in

[10] They are considered statistically significant when their p-value is less than 0.05.

this chapter is Fisher's Exact Test for Count Data. In our study, all the contingency tables were small enough to run Fisher's test in a reasonable time.

2.3.1 SM-RQ1. What are the demographic characteristics of the studies about OSSECOs?

2.3.1.1 SM-RQ1.1 In which type of sources are articles mostly published?

Figure 4 in Section 2.2.1 shows the distribution of the primary studies per publication type. In this section, to analyze this distribution, we compare it with that of the general context of publications in software ecosystems. In order to do this, we used the results reported in the secondary studies: [38], [37], [39], [49]. Figure 14 shows that the percentage of publications in journals is quite significant, and most of them are from journals with high impact factors such as IST and JSS (i.e., 17 studies from [39] and 6 studies from these studies are from these journals). Similar to [39], we think that this is beneficial for the maturity of the SECO and OSSECO field.

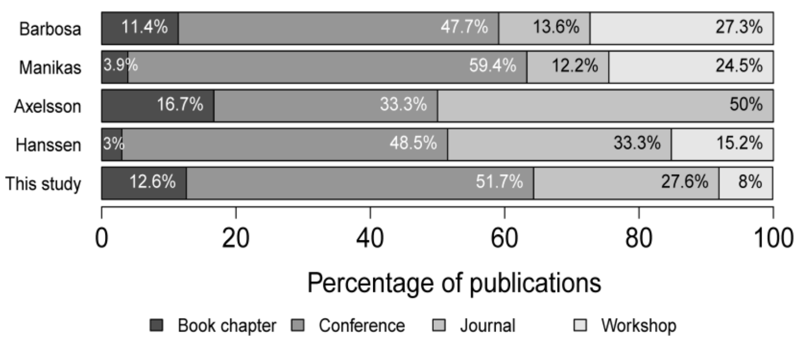

Figure 14. Comparison of types of publications.

2.3.1.2 SM-RQ1.2 How has the number of publications evolved over the years?

OSSECO is a growing research area in software engineering. Figure 15 shows a significant increasing trend in the number of publications related to OSSECOs with 56.3% out of the 87 papers studying and analyzing OSSECOs. Furthermore, since 2006 there has been a regular increase in the number of publications each year, with the exception of years 2012 and 2015, which does not significantly affect the overall trend. In addition, we have witnessed the emergence of a research community that shares interest in OSSECOs: IWSECO is an international workshop on SECOs with several publications on OSSECOs (e.g., [84], [85], [86], [87]), tutorials in relevant conferences like ICSE ([88]), specialized workshops such as WEA (workshop on software ecosystem architectures), and special issues about SECOs[11] in journals (e.g., IST[12], JSS[13]).

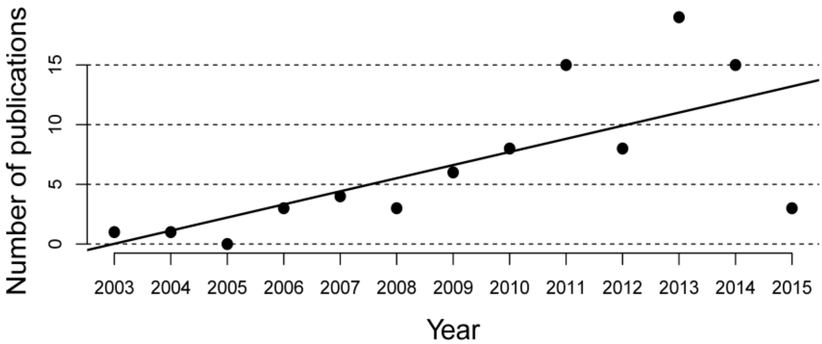

Figure 15. Publication trend.

2.3.1.3 SM-RQ1.3 How are papers geographically distributed?

We have put the results of our study in a general context of publications. In this case we use the context of computing science in the period 2006-2015 as presented in the bibliometric indicator database of the SCImago journal & country rank [89] (see Figure 16). It is no surprise that European and North American authors are the dominant researchers. However, in our study, the percentage of publications from Europe is significantly higher than in the SCImago database (67.8% and 35.7%, respectively). This could be due the increasing research on the OSSECO topic in some countries (e.g., The Netherlands and Belgium). On the other hand, the number of publications from Asia is surprisingly and significantly lower (38.7% in the SCImago database and 5.8% in our study).

In the distribution of papers in Europe, The Netherlands and Belgium are the countries with the most publications (25.4% and 13.6%, respectively). These values are corroborated by the countries of affiliation of the dominant

11 Several papers of these special issues are about OSS

12 http://www.sciencedirect.com/science/journal/ 09505849/56/11

13 http://www.sciencedirect.com/science/journal/01641212/85/7

researchers. This highlights the fact that in a relative, new discipline such as OSSECOs, leading research groups can create predominant niches in a specific research area, as it happened with Jansen and Brinkkemper's research group from the The Netherlands and Mens and Goemine's research group from Belgium.

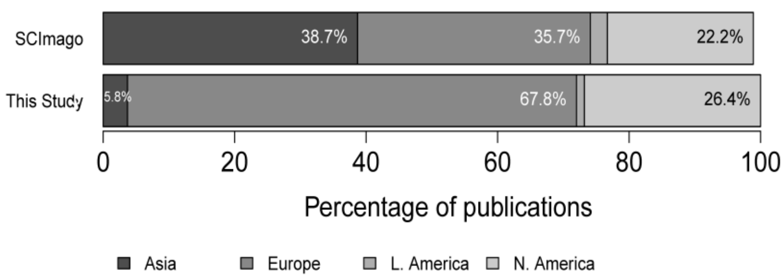

Figure 16. Comparison of publications per continent.

2.3.1.4 SM-RQ1.4 Who are the predominant researchers?

Table 13 shows that five of the predominant researchers are included in all of the top 10 lists. Brinkkemper, Herbsleb, Jansen, Mens and Goeminne are key entities in the social networks (i.e., keystone actors, network brokers, etc.). This is due to their strategic position in the social networks of authors. The measures used in this work highlight the authors connecting dispersed partitions of the OSSECOs researchers. Thus, we can identify that there are clusters (i.e., sets of authors collaborating together) around the main researchers. This would mean that amount of the research on the OSSECO topic is growing around these authors and their approaches. Also, this cluster enables independent authors to come together as a larger social network of shared knowledge about OSSECOs.

Some authors like Jansen, Mens and Goeminne have several publications about OSSECOs (i.e., 17, 9, and 7, respectively). This may explain the clusters around them. The number to citations to these publications explains the high values of their measures in Table 13. In contrast, other authors like Herbsleb and German, with high values in Table 13, are not in main clusters because they have only two papers in our set of primary studies. On the other hand, Lungu is on one of the predominant authors lists because of the number of references to his publications (23). Also, he is in one of the main research clusters because he is a coauthor of Lanza, who is one of the main nodes in his cluster.

Graph density has a value between 0 and 1 and describes how interconnected a network is [79]. Table 14 shows that the author-coauthor network density is very low (only 1.6%). This suggests that most of the authors only have a high-density relationship only with a small number of other authors. It is also an indicator of low network cohesion and membership. Other studies about the measures of author-coauthor networks have similar density measure result (e.g., [90], [91], [92], [93]).

This could be due to the youth of the field, and it could mean that it is a challenge to grow partnerships inside the OSSECO research community. In addition, it is necessary to find brokers that connect dispersed clusters.

2.3.1.5 SM-RQ1.5 How are publications distributed between academy and industry?

Figure 17 shows that in the period 2006-2016 for both the context of publications in computing science [94][14] and for this mapping study, the great majority of the papers are from academy. It is no surprise that academics are clearly more motivated to submit papers to journals and conferences. This is particularly true in the OSSECO domain where researchers are more interested in abstract concepts and definitions than practitioners, who are more attracted by practical questions. However, the number of papers from industry indicates that OSSECO is a topic of interest from the industrial perspective. We found a correlation between Ecosystem definitions and Type of papers: 53 (66%) out of the total of 80 papers that use ecosystem definitions are from academy (p=0.031).

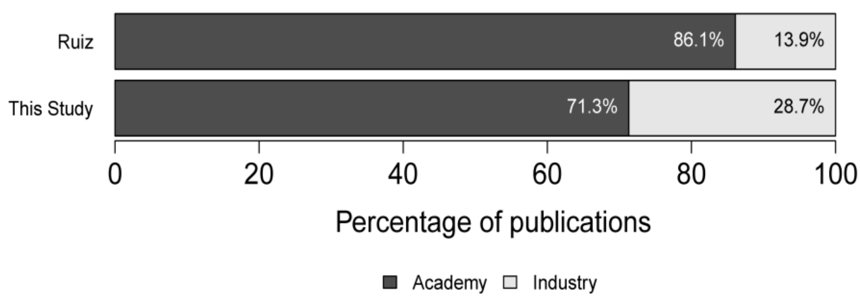

Figure 17. Comparison of affiliation.

2.3.1.6 SM-RQ1.6 What type of papers are published?

To analyze the distribution of the type of papers found in our study, we contextualized our results with Hauge et al. [71], which is a SLR about adoption of OSS in software-intensive organizations. Figure 18 shows that there are no remarkable differences in the distribution of papers between our study and Hauge et al.'s study.

[14] To obtain these values we contacted the author of the paper and asked him to provide the updated information.

This is an interesting fact because each type of paper contributes differently to the research community [95]. While non-empirical studies help to develop concepts and build theory, empirical studies provide concrete evidence for testing theories. For instance, on the non-empirical study side, we have: [R29], which describes the use of active theory in OSSECOs; and [R35], which proposes a conceptual framework to evaluate OSSECO's health. On the other hand, on the empirical study side we have: [R48], which makes a survey on SECO governance; and [R36], which makes a survey on SECO associated models. Figure 19 shows that neither of the top continents, Europe and North America, have more industrial papers than academic ones. Finally, experience reports provide examples of the use of theories in this side, we have: [R39, R42], which visualize the GNOME dynamism; and [R4], which shows the OSAMI-Commons project that defines a cross platform of an open-service ecosystem.

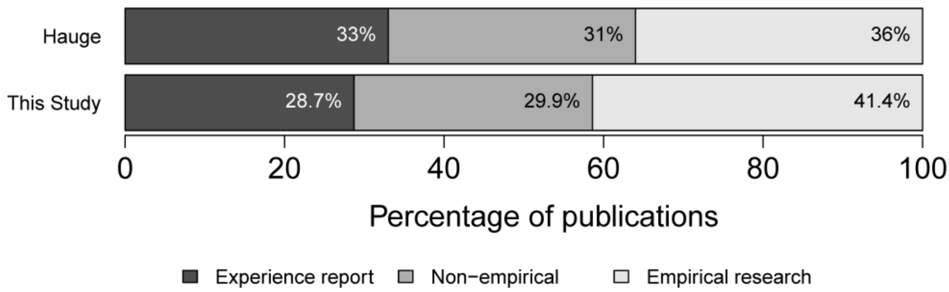

Figure 18. Comparison with respect to the type of research.

In contrast to other mapping studies, we did not find a correlation with $p<0.05$ between types of papers and continents. However, we did find a correlation between Publication year and Paper type. The number of empirical research papers has been increasing (4 between 2003-2008 and 30 between 2009-2015) ($p=0.025$). This can be interpreted as a sign of increasing maturity of the OSSECO field [39].

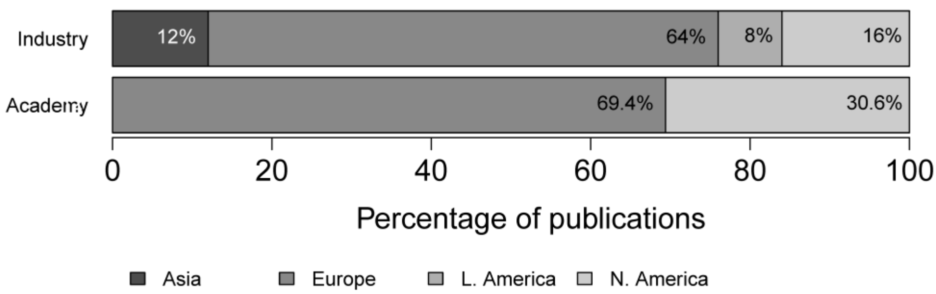

Figure 19. Affiliation per continent.

2.3.2 SM-RQ2. What is an OSSECO?

2.3.2.1 SM-RQ2.1 What definitions are related to the OSSECO definition?

To discuss this SM-RQ, we split this section into two parts. In the first part, we define a genealogical tree of the definitions that are related to the OSSECO definition. In the second part, we analyze the common elements across these definitions.

OSSECO related definition evolution.

In order to clarify the relationships between the definitions related to OSSECOs and to contribute to the understanding of the OSSECO phenomenon, we wanted to picture their chronological evolution. Moreover, we attempted to depict the research in this field that we found in our mapping study. Thus, we built a genealogical tree with the ecosystem definitions, their relationships, and their predominance in the OSSECO community (see Figure 20). The figure can be read as follows: (a) from left to right, the figure shows the evolution of the OSSECO definition over time; (b) from right to left, the figure shows the inheritance relationships between the different ecosystem authors definitions; (c) from top to bottom, the figure shows the evolution of each ecosystem definition; and, (d) each node in the figure shows the first author and the number of citations per publication. This number corresponds to the papers (from our set of primary studies) that cited that publication for the definition used in their research work. This means that references for other purposes were not considered.

Figure 20 shows that SECO is the most frequently referenced definition in our set of primary studies. Furthermore, there are several references to the BECO definition. Wynn [R34] references the BECO and DBECO articles in his paper and

Jansen et al. [R40] references the SECO, BECO and DEBECO definitions. This indicates that the OSSECO is a specialization of these definitions.

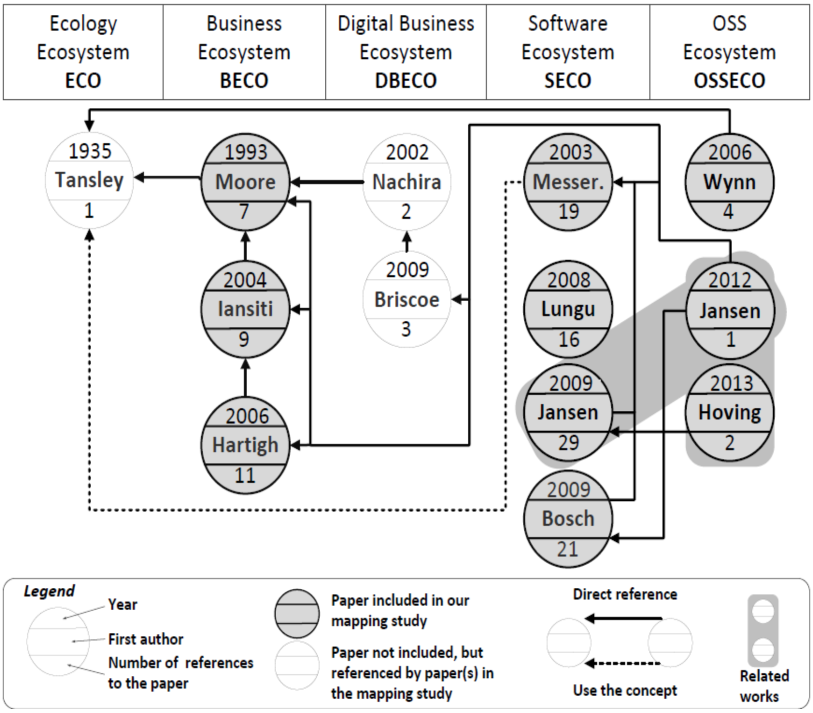

Figure 20. Evolution of OSSECO definition.

The genealogical tree shows that there are several direct and indirect relationships among all of the definitions related to OSSECO. In our systematic mapping, we found few papers that try to adapt ecosystem theories to the OSSECO domain (i.e., [R34] uses the entropy concept and [R82] uses the predators and prey concept). The rest of the papers simply use the ecosystem definitions (i.e., BECO, DBECO, SECO, OSSECO) to identify the actors, the relationships, and the specific environment of a specific OSSECO (e.g., [R8] for Nagios, [R12, R87] for Ruby, [R17] for Eclipse). Furthermore, we found three papers that use the health metaphor to analyze OSSECOs (i.e., [R52, R56, R62]). However, similarly to [69], we did not find the application of theories, models, or ideas from ECOs to the domain of OSSECOs, despite the fact that ECOs have been studied for many decades. It is a challenge for OSSECO researchers to transpose theories and ideas from ECOs (e.g., systems dynamics modelling, general system theory) to OSSECOs.

Common elements.

Figure 21 shows a conceptual map that represents the relationships between the five OSSECO related definitions and their terms. We found that there are common elements across definitions:

- A community of actors (i.e., complex organisms in ECO, business world organisms in BECOs and DBECOs, and collections of products, projects, software solutions, and businesses in SECOs and OSSECOs).
- A set of relationships.
- An environment (i.e., economic communities in BECOs, open socio-technical systems in DBECOs, shared market and technological platforms in SECOs and OSSECOs).

The ecosystem metaphor is useful for explaining the dynamics of complex systems such as business, digital, and software systems. The software ecosystem metaphor was coined 13 years ago, by Messerschmitt and Szyperski [96], reflecting and incorporating software technology into BECO. However, we only found one study that discusses the metaphor in depth [69]. Most of the papers have only adopted common definitions of SECO or related definitions (see Section 2.2.2). In our opinion, in the near future, most SECOs, BECOs, and DBECOs will be more open to become closer to OSSECOs and share some of their features. This is because SECOs are strongly related to BECOs and DBECOs and openness is not only a desirable characteristic of SECOs but a vital characteristic as well. Furthermore, every software platform at the center of an ecosystem has to have some degree of openness [33].

Finally, we find that there is currently a consensus among SECO researchers for two SECO definitions: business-centric definition of Jansen et al. [25] and the platform-centric definition of Lungu [27]. In our opinion, a commonly accepted definition of SECO is important in order to improve the communication between SECO researchers and practitioners and thereby reduce the subjective and ambiguous notions of SECOs.

2.3.2.2 SM-RQ2.2 Are there specific definitions of OSSECO?

OSSECOs are understood from two perspectives: (1) an ecosystem perspective, where OSSECOs are a network of actors, organizations and companies with symbiotic relationships that can be studied from a business-goal point of view; (2) a project-community perspective that focuses on technical and social aspects of a set of software projects and their communities [R53, R68]. We found that the three main authors of the clusters study OSSECOs from a project community perspective. However, in their most recent work, they make a call to action for future research in OSSECOs from an ecosystem perspective [R53, R56]. Table 19 shows the classification of the three-main author-coauthor clusters and the two OSSECO perspectives.

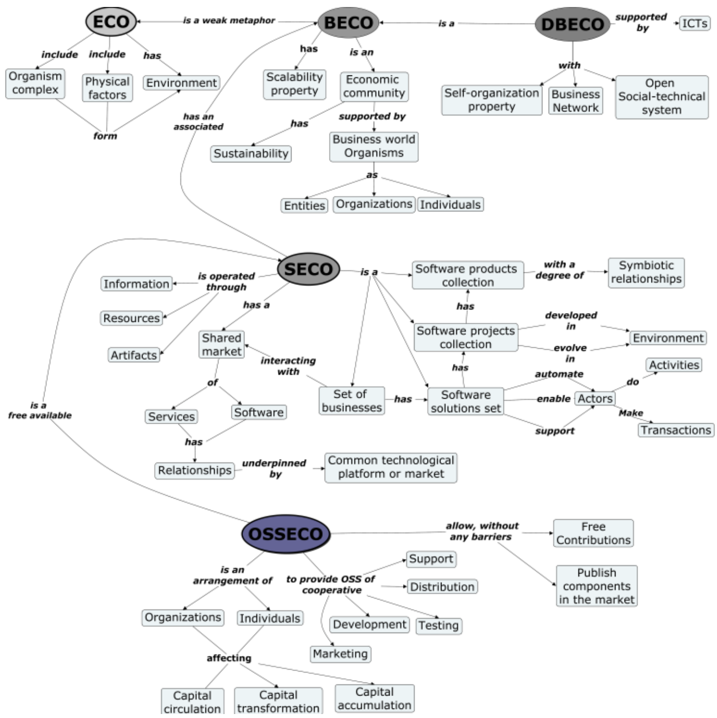

Figure 21. Conceptual map of OSSECO.

Some authors argue that OSSECOs are probably the most complex type of SECO [R74]. However, we found very few definitions of OSSECO in our study. Specifically, in the definitions of OSSECO (Table 15), we found the following as common elements:

- A set of heterogeneous units (e.g., organizations, software projects and services).
- Symbiotic relationships among units (e.g., capital, projects, components).
- An open perspective in a shared market (e.g., to provide support to OSS, to add contributions without barriers, to provide freely available results for everyone)

We did not find any explicit reference to OSSECO communities in the definitions. This is surprising because the OSSECO communities are one of the most important differentiators between OSSECOs and other types of ecosystems. Finally, we distilled an OSSECO definition (see Section 2.5.1) because the three above-mentioned definitions (see Table 15) have a lack of specificity in the particularities of OSSECOs (i.e., OSS community, open-common platform).

Table 19. Classifications of authors clusters.

Cluster	Ecosystem perspect	Community perspective
Jansen et al.	Implements frameworks for OSSECOs health measuring [R62].	Provides studies of data repositories of OSSECOs like Ruby and Debian [R12, R30]. Gives a set of models for ecosystem governance and OSSECO enterprise [R36, R41].
Mens et al.	Provides a framework for analyzing OSSECO communities [R53].	Implements tools for modelling OSSECO communities and projects [R10, R19]
Lanza et al.	Provides a view of SECOs as a collection of software projects developed within and across organizational boundaries[R39].	Implements tools for visualizing OSSECO projects [R39]

2.3.2.3 SM-RQ2.3 What elements belong to an OSSECO?

In order to validate our manual content analysis approach to collect data for answering SM-RQ2.3 (see Section 2.2.2), we compared its outcome (65 terms belonging to the OSSECOs identified) to that of a computational approach. To this end, we used a text-mining approach based on co-occurrence and term frequency analysis as defined by Salton and Buckley [97]. In their work, the content of a document is represented as a vector space, i.e., $D = (w_1, w_2, \dots w_k)$ where w_k represents the weight of term k in document D that is calculated upon the term occurrences (tf) and the inverse document frequency (idf)[15]. This method allowed us to identify the importance of each term in the corpus[16]. Different terms have different importance in a text, and so w_i is an indicator that represents how much the term t_i contributes to the semantics of document D. This approach is different from the one described in Manikas [39], who identifies the keywords of the set of papers. However, he took these words from the keyword field of each paper. We are taking the terms from the entire text of the paper.

In order to get w_i and compare our terms with the most weighted terms in the corpus, we used the R text mining package [98] and followed the steps from Narang [99]: (a) we obtained a document term matrix of 23617 columns (i.e., terms) and 87 rows (i.e., documents); (b) we calculated the weight for all terms in the document term matrix as defined by Salton and Buckley (i.e., $tf_{i,j}.idf_i$); (c) we

[15] The idf varies inversely with the number of documents N to which a term is assigned.

[16] The corpus is a set of documents on which to perform the text analysis.

sorted the list of terms by weight; and (d) we searched the position of each of the 64 terms found in Section 2.2.2 on the list of sort weighted terms. Table 20 shows the distribution of the number of OSSECO terms across the weighted interval ranking.

Table 20. Distribution of OSSECO terms.

Interval rank	OSSECO terms	%	Acum
1-100	31	48\%	48\%
101-200	9	14\%	62\%
201-300	5	8\%	70\%
301-400	3	5\%	75\%
401-500	1	2\%	77\%
1000-2000	5	8\%	85\%
2001-5000	4	6\%	91\%
upper 5000	3	9\%	100\%

Table 20 shows that 48% of the OSSECO terms that we manually identified appear in the top 100 of the weight matrix terms. It also reveals that 77% of these terms are among the 500 most ranked terms in the corpus. This may indicate that the use of well-known terms is significant in the OSSECO research community. We found that 70 terms in the top 100 of the weight matrix terms do not appear on our list of the 64 terms identified. The reason is that they are mostly common terms in the software engineering domain or general words (e.g., syntax, error, analysis, software, systemic, component, etc.).

In order to analyze the OSSECO terms below the rank of 500 (e.g., wishlist, vocabulary, entropy, bug reporter, sub community, adopter, IRC, bug fixer, and passive user), we calculated the keyness[17] of the 64 OSSECO terms. To this end, we used the Scott and Tribble approach [100] to calculate keyness using log-likelihood tests. This is a statistical function used for comparing word frequencies of linguistic features in two or more corpora [101]. In this work, the OSSECO corpus is the sub-corpus, and the corpus academic vocabulary list of contemporary academic English (consisting of 190.000 documents) was used as a reference corpus [102].

We found that there is a large disparity in values. The term with the highest keyness value is project (31694) and the one with the lowest keyness is super repository (33.76). All of the terms in the group with the lowest weight are among the 20 ones with the lowest keyness. However, all of the OSSECO terms are positively key, meaning that they occur more often than would be expected by

[17] Keyness is a term used in linguistics to describe the quality a word or phrase has of being key in its context. Keywords are items of unusual frequency in a given sub-corpus in comparison with a reference corpus Scott (1997).

chance in comparison with the reference corpus [103]. In other words, the OSSECO research community tends to overuse the terms related to OSSECOs more than the academic community in general. This may indicate that the research community is able to create a common vocabulary, which could represent a first step towards an ontology of OSSECOs. In Section 2.5.2, we present a taxonomy as a first step towards such an OSSECO ontology. We think that the OSSECO ontology is necessary in order to allow semantic interoperability between the distributed and heterogeneous OSSECO actors.

2.3.2.4 SM-RQ2.4 What instances of OSSECOs have been reported in the literature?

Table 21 shows that most of the papers found in this review are about OSSECO instances. This is because OSSECOs have several kinds of data sources such as: project sites, ecosystem hubs, and aggregation sites [104]. These data sources are freely available and tend to contain the entire history of all OSSECO projects, community relationships, and their artefacts. In addition, OSSECO researchers also use and develop dedicated tools to get a better insight into how the ecosystem surrounding an OSS project affects its evolution [105]. We can conclude that, because of the openness of the OSSECO repositories, they are ideal for statistical and network analysis research.

In our study, Eclipse was the predominant OSSECO studied. It was analyzed from different perspectives (e.g., OSSECO licensing models [R9], co-creation process in OSSECOs [R17], global SECOs [R28], OSSECO co-evolution [R43] and OSSECOs marketplaces [R68]). We compared our result with two previous mapping studies (i.e., Manikas and Hansen [65], and Ameller et al. [70]). In their work, Eclipse was the most referenced OSSECO, 16% and 41.7% respectively. Eclipse's popularity among researchers may be due to the less restrictive Eclipse Public License [R1], the Eclipse incubation programs [R13], the common development infrastructure, the possibilities of co-creation and co-evolution with relevant partners, among other important aspects.

2.3.3 SM-RQ3. Which representations have been proposed for OSSECOs?

2.3.3.1 SM-RQ3.1 Which primary studies use models to represent OSSECOs?

Figure 22 shows the numbers and percentage of papers that use OSSECO models in the secondary studies of Hanssen and Dybå [38]; Barbosa et al. [37]; Manikas [39]; Axelsson and Skoglund [49], and this study. Significant differences in the five studies can be observed. However, this might be due to the fact that we used a more flexible criterion for paper classification (i.e., we selected a paper if it

Table 21. SECOs instances and papers related.

OSSECO	Papers
Eclipse	R9, R17, R28, R31, R34, R37, R40, R41, R43, R44, R48, R67, R68, R69, R73, R36
GNOME	R10, R11, R19, R39, R42, R53, R60, R62, R64, R76
Android	R48, R70, R86
Ubuntu	R16, R48, R50
Ruby	R12, R48, R87
Open Design Alliance ODA	R36, R41
Debian	R16, R30
Python	R62, R65
Wordpress	R48, R51
Brazilian Public Software (BPS)	R2, R88
Apache	R85
Ecos	R70
Evergreen	R35
FOSS4G	R22
Gurux Software	R13
Moodbile	R84
Nagios	R8
NASA Earth science	R15
OSAMI Consortium	R4
OSGeo	R25
OSMOSOFT	R47
OpenStack	R77
R	R67
Topcased	R9
Vaadin	R13
Webkit	R58

had any OSSECO model). In contrast, Manikas and Barbosa were more restrictive in their criteria (i.e., Manikas selected papers with empirical and analytical models, Barbosa selected papers with software product line development models and OSS development models, all of the papers from Axelsson have qualitative or descriptive models, and Hanssen identified papers describing and modelling ecosystems). Nevertheless, we agree with Manikas. when he argues that there is a lack of papers using models based on automatic or mathematical manipulation for solving a specific problem and there is an excess of papers using ad hoc models.

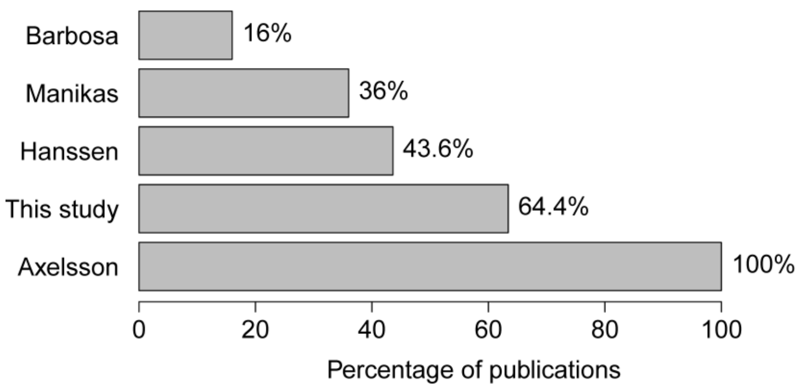

Figure 22. Comparison of publications with SECO models.

2.3.3.2 SM-RQ3.2 Which of the proposed models, if any, are specific for OSSECOs?

OSSECO modelling has emerged as an important research area in software engineering [106]. In our literature review we identified several specific OSSECO models and meta-models to describe and analyze the complex relationships between members in specific OSSECO case studies. However, there is no unified model language for OSSECOs. We found that researchers of OSSECOs used several types of modelling techniques that are specifically adapted for only one or a few research studies. Nonetheless, there is still a need for modelling OSSECOs due to the following: (a) Complexity- Since SECOs have several types of actors, resources, implicit boundaries, shared market, licenses issues, etc.; they are complex artifacts [69] and we need to understand them; (b) Traceability- Since the software industry is constantly evolving and is currently undergoing rapid changes [107], it is important to understand OSSECO evolution by analyzing its historical data sources, and (c) Communication- Because of the complex network of symbiotic relationships between entire social actors, open source communities and commercial software companies, etc. [108], the heterogeneity of OSSECO stakeholders will require a common language to facilitate communication. In other words, OSSECO modelling needs to be complemented by more research efforts that focus on providing model-based approaches to describe and analyze OSSECOs.

2.3.3.3 SM-RQ3.3 Which notation and guidelines have been used for modelling OSSECOs?

We found several notations for modelling OSSECOs. However, all of them adapt available modelling techniques or use ad hoc models to support their works without proposing new modelling techniques. We think that the development of new modelling techniques for OSSECOs is important because it has evolved from different domains (i.e., ecosystem and BECO). These domains are not directly related to the software engineering base of knowledge nor have the software modelling techniques been designed for the complex elements and symbiotic relationships of software ecosystems. Furthermore, there is a small but growing line of recent research efforts that is specifically focused on providing model-based approaches to describe and analyze SECOs [109]. These conditions are necessary for modelling OSSECOs in a systematic way. In addition, they allow abstracting and reasoning about OSSECOs [110]. Table 22 shows the contingency table for the type of ecosystem and models used.

Table 22. Contingency table for ecosystem and models.

	BECO	DBECO	ECO	OSSECO	SECO
Ad hoc	8 (14.5%)	1 (1.8%)	0 (0%)	0 (0%)	9 (16.4%)
Class Diagram	1 (1.8%)	0 (0%)	0 (0%)	1 (1.8%)	2 (3.6%)
Conceptual Map	2 (3.6%)	1 (1.8%)	0 (0%)	1 (1.8%)	1 (1.8%)
iStar	3 (5.5%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Mathematical	1 (1.8%)	0 (1.8%)	2 (3.6%)	0 (0%)	2 (3.6%)
Metamodel	1 (1.8%)	0 (0%)	0 (0%)	0 (0%)	3 (5.5%)
SNA	1 (1.8%)	0 (0%)	1 (1.8%)	0 (0%)	5 (9.1%)
Tabular	2 (3.6%)	0 (0%)	0 (0%)	4 (7.3%)	3 (5.5%)

2.3.3.4 SM-RQ3.4 What type of analysis was conducted using the models identified in SM-RQ3.3?

Jansen et al. [83] identify three important uses of SECO modelling, one of which is SECO analysis. However, we found that most of the papers that use models have not conducted any OSSECO analysis (see Figure 13). In addition, the analysis techniques used in the remaining papers, such as mathematical, visual, statistical, and SNA techniques were used to analyze specific cases. They are insufficient when a more in-depth analysis is necessary. In agreement with other authors (Barbosa et al. [37]; Jansen et al. [83]; Manikas [39]), we think that developing analysis and modelling techniques is one of the most important challenges in the OSSECO domain. We found a correlation between Type of analysis and Model type: 32.1% (18) of the papers that conducted some type of analysis use adhoc models.

2.4 Threats to validity

As in every empirical study, there are several threats that might negatively affect the validity of this systematic mapping. In the protocol, we identified and tried to mitigate them using four categories: construct validity, internal validity, external validity, and conclusion validity (see [111] for details of this classification).

2.4.1 Construct validity

The construct validity category includes three major threats. The first threat is that the research questions may not cover all the relevant aspects that characterize the existing research in our area of interest. To minimize this risk, we used a brainstorming technique with the participation of all the authors of the study to define them. The second threat is that the inclusion of all the relevant works in the field is not guaranteed. This threat was mitigated by combining several databases and manual searches to selected journals and conferences from previous literature reviews on software engineering and OSS. However, this issue may not have been solved since the problem goes beyond an accurate protocol and also concerns issues related to the paper (e.g., inaccurate abstracts). To mitigate this risk, we included the papers from two other literature reviews ([38], [37], [39], [49]) and all of the chapters of the only existing book that is centered on the study of SECOs [112]. Finally, there is a risk of obtaining a biased selection. To mitigate this risk, inclusion and exclusion criteria guided the selection, and a multi-stage process involving more than one researcher for each paper was used to perform it.

2.4.2 Internal validity

There are two threats to internal validity in this systematic mapping. The first threat is that most of the papers do not provide accurate definitions or references for the OSSECO term. For instance, several papers use definitions related to SECOs and they study OSS communities or OSS projects (see Figure 8 and Table 21). The second threat is related to the identification of values for classification criteria: for some of the criteria to classify the papers, the possible values were not obvious with regard to OSSECO related definitions, one author identified the possible values, and the list of definitions was discussed and analyzed closely by all of the authors of the paper. Furthermore, we calculated a word frequency table from the documents and added other 16 new OSSECO related definitions. With regard to OSSECO models, we found a lack of modelling techniques to represent OSSECOs. We decided to identify the different ecosystem definitions of the authors and classify the techniques and notations used in each paper to model ecosystems. This process minimized the risk because several papers use these techniques to model OSS communities or OSS projects.

2.4.3 External validity

Since our results are within the scope of OSSECOs and we do not attempt to generalize conclusions beyond this scope, external validity threats do not apply.

2.4.4 Conclusion validity

Conclusion validity is concerned about whether the research performed is reproducible by other researchers with similar results. In this regard, we have explicitly described all of the steps performed in the systematic mapping by detailing the procedure as defined in the research method (See Section 2.1.1). We have also created an online document with details that are not central to the paper but that are necessary to ensure reproducibility and provide evidence about our findings.

2.5 Further work

The analysis of the results allows us to state that OSSECO is a growing research area in software engineering [R16, R49, R50]. Due to this, there are several new research opportunities in the empirical examination, modelling, analysis, measuring, quality evaluation, etc. of OSSECOs. Along with this argumentation, in this section we provide two initial proposals to improve the current structure of the knowledge on OSSECOs: a definition for OSSECOs and a taxonomy of OSSECO related terms.

2.5.1 The OSSECO definition

In any domain, the concept of ecosystem can be difficult to define clearly. This is true even among scholars in ecology, its native discipline. According to our study, there is a relation between BECOs, SECOs, and OSSECOs (see Figure 20). By combining the definitions of SECO, BECO, and DBECO that we found in our mapping study, we define an OSSECO as: a SECO placed in a heterogeneous environment, whose boundary is a set of niche players and whose keystone player is an OSS community around a set of projects in an open-common platform. Table 22 details the OSSECO definition. The first column shows the breakdown of the OSSECO definition. The second column describes the definition related elements in an OSSECO. The third column references the source of the definition component. Finally, the last column contains specific examples.

Table 23. Breakdown of OSSECO definition.

OSSECO definition break down	Description	Source	Examples
a SECO placed in a heterogeneous environment	In OSSECO is an economical social and technical environment	Iansiti and Levin BECO	Other OSSECOs, commercial SECOs, Government, Market rules, synaptical relationships, etc.
whose boundary is a set of niche players	In OSSECO there is more than one.	Jansen et al. SECO	Partners, Re-sellers, Platform provider, etc.
and whose keystone player is an OSS community around a set of projects in a open-common platform	In OSSECO keystone players drive platform technologies and the standards (Aarnoutse et al. 2014).	Lungu et al. SECO	Contributors, passive users, data sources, etc.

2.5.2 Taxonomy of the OSSECOs terms

In this subsection, we present an initial taxonomy composed of the terms that we found in our review. In order to do this, we applied the second cycle of our data extraction process to the results from Section 2.2.1.6 We then grouped the OSSECO terms into three dimensions, which we had presented in a previous work [60]: (a) the software platform which groups the terms related to the technology or market around which the ecosystem is built; (b) the OSS community, which groups the terms related to the community (or set of communities) of the ecosystem; (c) the ecosystem network, which groups the terms related to the ecosystem as a network of elements, such as projects or companies. These categories are related to the SECO viewpoints defined in Section 2.2.2. In addition, we divided the categories into subcategories based on the categories from Bosch [23], the levels from Jansen et al. [25], and the dimensions from dos Santos and Werner [26]. This taxonomy, which is presented in Figure 23 aims to serve as the starting point for establishing a common terminology for OSSECO[18].

[18] In Figure 23, the references to Jansen et al. [25], dos Santos and Werner [26] and Bosch [23] are abbreviated with the name of the first author for the sake of brevity.

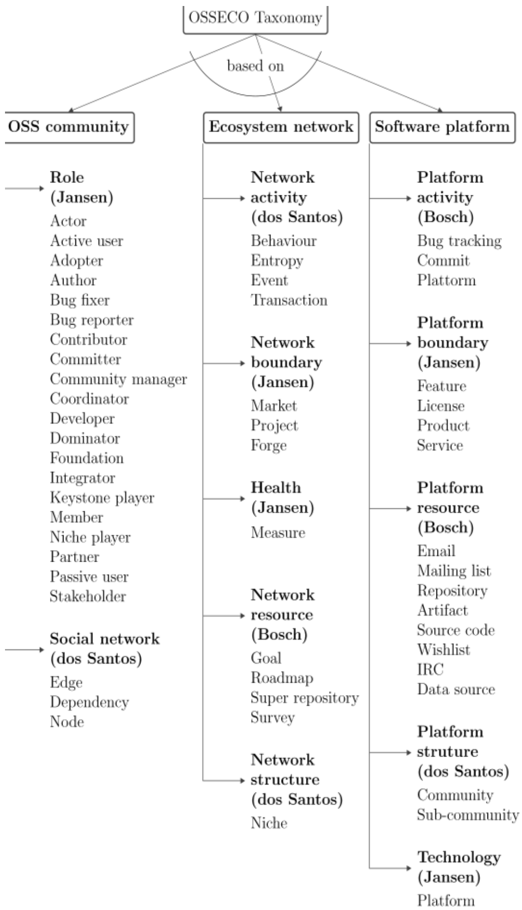

Figure 23. OSSECO taxonomy: an initial proposal.

Many of the terms in the taxonomy are not exclusive to OSSECOs; however, many of them exhibit some characteristics that are specific to the OSSECO domain. For example:

- A network boundary around an open or semi-open platform boundary has the potential for numerous benefits, including enhanced adopter offers through the use of innovation potential in the OSSECO [R1].
- Unlike to other software distribution paradigms, source code is usually available from OSSECO repositories. This facilitates some software quality practices like peer reviews [49].
- In OSSECOs, the relationships between keystone players (e.g., the OSS community) and niche players (e.g., partners, providers, adopters) are under an OSS license schema. It is sometimes difficult to control because there are

different licenses with specific characteristics that are not always compatible [51].

- In OSSECOs, the OSS community usually dominates the development instead of an individual organization (this could happen indirectly because sometimes a community is influenced by a single organization indirectly) [38]. The community defines a roadmap that guides the development.
- The OSSECOs typically provide access to all data repositories related to their evolution (i.e., how software changes over time) [12]. Also, another feature of software repositories is the option to fork or copy a whole OSS project and start a different forge of the project [39].

Table 24 shows a general description of each taxonomy term according to the primary studies. We are currently developing an OSSECO ontology based on the taxonomy presented here. This ontology is intended to support QuESo, a framework for the representation, synthesis, analysis, evaluation, and evolution of OSSECOs [20]. For this purpose, we are improving the OSSECO taxonomy by standardizing and extending the set of terms, and we are using OWL to describe the relationships among the concepts. Finally, we are defining a set of axioms and inference rules to represent the meaning of these concepts in a formal way to support reasoning.

2.6 Research roadmap for OSSECOs

In this section, we outline a research roadmap for OSSECOs. First, we compile the few studies in the broader area of SECOs that have identified research challenges. The first study in that direction was Jansen et al. [24], who mentioned several challenges, notably characterization and modelling of SECOs. Barbosa et al. [37] identified eight major fields within the software ecosystem domain; it is worth mentioning that one of them is the further study of OSSECOs. Hanssen and Dybå [38] uncovered several theoretical challenges about SECO, which are specifically related to socio-technical theory. Finally, Manikas [39] proposed two approaches to address complexity and theory building in SECOs. Table 25 summarizes these SECO challenges and the papers in our set of primary studies that addressed them in the context of OSSECOs.

The primary studies listed in Table 25 only provided partial answers to the fundamental questions behind these four challenges. Furthermore, some other aspects were not mentioned in the four papers on SECO challenges, but they do appear in some of the primary studies that we have surveyed. As a result of both observations, we outline the following research agenda:

Table 24. taxonomy terms.

Term	Description
Active User	Active users comprise occasional developers and users who report bugs, but do not fix them [R35].
Actor	Actors are either users or contributor [R29].
Adopter	Who do not contribute directly to the platform, but use it to develop tools [R31].
Artefact	The software project is defined as a structured collection of artefacts linked by derivations and produced to support/provide a collection of in use behaviours in order to satisfy a set of user requirements [R16].
Author	The author is the person that actually made the changes to the committed files [R19].
Behavior	No definition found in the primary papers.
Bug Fixer	Who Fixes reported bugs [R64].
Bug Reporter	Who Reports bugs [R64].
Bug Tracking	Track bug tracker activity (e.g. bugs opened, closed, statuses changed) bug tracker increase the source code centrality for a developer [R11].
Commit	These are pieces of atomic changes done on the source code [R10]. The developers' commits to the project reflect not only the technical contributions but also the social and collaborative aspect of those contribution [R25].
Committer	Who contribute directly to the platform making changes in the OSS data sources[R69].
Community	OSS community is a social ecosystem on its own and in junction with other OSS communities. However, it differs from other social networks in its hierarchical structure. [R13]. Set of individual and shared resources of people\textquotesingle s time, effort, attention, skill, sentiment (beliefs and values), and computing resources are part of the socio-technical web of FOSS [R23].
Community Manager	Who is the responsible of the OSS community governance [R41].
Configurator	No definition found in the primary papers.
Contributor	They contribute in some form to the OSS project [R20]. Contributors obtain private benefits from the development of shared assets that are not available to free riders, who only use the asset [R29].
Coordinator	No definition found in the primary papers.
Data Source	Code repository containing all versions of the source code, the bug tracker containing all feature requests and problem reports as well as all the resolution process, and the mailing list(s) containing all the mails exchanged among developers and between users and developer [R10, R53].
Dependency	It defines work interdependence among the ecosystem members [R3]. It is a symbiotic relationships between ecosystem actors [R38].
Developer	They contribute to OSS projects code for the personal gratification that comes from increasing their reputation among peers [R20]. They are primarily volunteers [R64].
Dominator	Is the actor that control the value capture and value creation of the ecosystem. [R56]
Edge	A-B in the network is created if an actor B replies to a message earlier sent by an actor A [R8]. Edges between ecosystem actors represent projects on which they collaborated [R14].
Email	No definition found in the primary papers.
Entropy	As a system is modified its disorder or entropy always increase. This is know as software entropy [R34].
Event	OSSECO organized events where stakeholders are brought together that share an interest in the total ecosystem [R62].
Feature	It identifies new functionality and enable develop the software in a common and creative way [R1].
Forge	Are Open Source Software (OSS) repositories designed to support teams doing software [R26]
Foundation	It is a democratic model based on voting rights, or a benign dictatorship (such as the Linux kernel), leadership will bean extremely important aspect of the ecosystem\textquotesingle s development [R5]. Foundations provide financial, organizational, and legal support to the broader free in OSS [R25]. This economic community produces goods and services of value to customers, who are themselves members of the OSSSECO [R27].
Goal	The goal was to provide stakeholders in OSSECO with insight into their ecosystem development and the most important metrics that indicate success in these ecosystems [R62].
Integrator	System integrators deliver solutions by selling a stack of hardware, software, and services as one product [R20].
IRC	It is a real-time chat [R11]
Keystone player	A keystone player is an actor in the ecosystem, whose contribution to the ecosystem stimulates the health ofthe entire ecosystem [R41].
License	OSS may be defined as software released under the terms of a license that basically allows the licensee to use, modify, and redistribute, either gratis or for a fee [R1].
Mailing List	It Contain all the mails exchanged among developers and between users and developer [R10].
Market	It is a phenomenon that occurs when the good is a shared resource such as a file format or software platform [R27]. The market as a regular player in a software ecosystem, assuming it plays a role similar to that of other players, such as developers and user. The market as the ecosystem\textquotesingle s energy source, arguing that it plays a significantly different role from other players. It can directly or indirectly affect other players and determine the success of a software product [R68].
Measure	It is an indicator for OSS community health.
Member	It can start by directly contributing to code without prior socialization [R11]. It take part in the OSS community membership program [R36]. Customers, who are themselves members of the ecosystem, the member organisms also include suppliers, lead producers, competitors, and other stakeholders. [R48].
Niche	The software ecosystem niche in which a given OSSECO lies [R24] . The software supply networks that reveal which software ecosystem instances (or niches) each system exists within [R24].
Niche Player	Usually form the main volume of the ecosystem actors drawing value from the keystones. A niche player aims to separate from the other niche players by developing special functions [R56].
Node	Actors as nodes, tied or connected by one or more specific types of interdependencies [R25].
Partner	There are naturally business partners, industrial partners and similar interest groups participating outside the range of the model that are an integral part of an OSSECO [R13].
Passive User	Passive users are all remaining users who just use the system [R58].
Platform	It is set of software and services [R4] typically managed by an OSS community [R44].
Product	A product is a set of software intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [R14].
Project	OSS projects are typical environments in which SECOs develop around the community [R12].
Repository	It is a system which keeps and manages source codes [R17].
Roadmap	It defines planning and time-to-market (or more strictly time-to-technology) [R41].
Role	They representing the interaction mechanisms between the various actors that constitute a software system [R3]. Important roles in OSS are users, developers, core developers and project leaders [R8].
Service	These services are provided, the organization benefits from making explicit and sharing knowledge with partners, since the knowledge does not need to be made explicit when the organization provides these services again. [R46].
Social Network	Network derived from the data sources of an OSS project [R8].
Source Code	No definition found in the primary papers.
Stakeholder	There are three main groups of stakeholders: the publishing entity with its allocated resources for the project, the industrial partners and theirs developers, and finally existing opensource communities and other individuals [R13].
Sub Community	This OSS community may be subdivided in (possibly overlapping) sub-communities. For example, one can distinguish between the user community, containing all individuals who use an executable version of the software system, and the developer community, containing all individuals who are in charge of maintaining and improving this software system over time [R53].
Super Repository	It represents a collection of version control repositories of the projects of an OSSECO [R14].
Survey	No definition found in the primary papers.
Transaction	Transaction may be finished over a period of minutes, hours, or even days – thus the term, from a computational perspective, of long-lived or long-running transaction [R27].
Wishlist	No definition found in the primary papers.

- OSSECO modelling and analysis. Different authors mention the need to tackle the lack of a universally accepted set of modelling methods because this is hampering the advancement of software ecosystem research ([37], [109]). In addition, modelling large networks, scalable model visualization tools, and the study of ecosystem evolution and dynamics are some of the major challenges within the SECO domain [83].
- Socio-technical theory. The field of SECOs is missing an established theoretical background [39]. Socio-technical theory addresses important concepts such as organizational control, ecosystem self-regulation, network organization, the role of technology, and the sharing of values. These concepts are relevant in order to understand OSSECOs as the interplay between the social system and the technical system [38].
- Ecosystem knowledge. OSSECO data sources provide access to a variety of information about OSSECO evolution. However, information about social behavior in the ecosystem must be taking into account. In order to do this, machine learning, text mining, case-based reasoning, and other techniques [R65, R57] can be used to identify social issues such as no implicit relationships, community sentiment analysis, cross-references between OSSECOs, among many other challenges [R23, R85].
- OSSECO quality. The quality of OSSECOs affects organizations, adopters, software developers and the OSSECO itself. However, quality management and operationalization of software ecosystems is still an immature discipline. In addition, OSSECO quality is quite different from the standard ones (e.g., ISO/IEC 25010, in terms of production process, community, distribution methods, license types, social organization, support, etc. [48]. Therefore, OSS quality models emerged due to the inability of traditional quality models to measure these unique OSS features [40]. These quality models in OSS projects can be the basis of OSSECO quality models [50].
- OSSECO monitoring. The assessment of OSSECO health is usually realized by tools for a specific community or a specific platform. For instance, there are several solutions in the literature for the monitoring and analysis of specific OSS communities by accessing their available data repositories directly [R1, R4, R13, R83]. There is a need to implement frameworks that are able to: (a) monitor a list of OSSECO quality sub-characteristics over time; (b) link the gathered values with adopter needs by operationalizing quality requirements; and (c) engineer a portfolio of web services that support OSSECOs.

Table 25. SECO challenges.

Authors	Challenge	Addressed by
Jansen et al. (2009a)	• Characterization and modelling of SECOs • Developing Policies and strategies within SECOs for SECO orchestration • Determining a strategy to thrive and make profit in an SSN	R2, R10, R16, R87, R4, R47, R49, R56, R88, R13, R18, R20
Barbosa et al. (2013)	• How quality can be measured per developer How relationships are formed between developers • How conflicts are solved in OSS ecosystems • How decisions are made in SECO and how can be measured in code changes • How APIs to third-party component are used	R19, R51, R61, R39, R59, R63, R85, R86, R87, R40, R73, R53, R55, R64, R50, R70
Hanssen and Dyb°a (2012)	• Socio-technical theory • Related theory of organizational ecology	R37, R54, R77, R86, R87, R34, R38, R76

2.7 Conclusions

This chapter has reported a systematic mapping in the field of OSSECOs with the goal of identifying and examining the state of the art on this topic. We designed and followed a rigorous protocol, which uncovered up to 87 papers from a gross total of 683, to answer the different research questions that we identified. We may consider the answers to these questions as the main outcome of this study.

2.7.1 SM-RQ1. What are the demographic characteristics of the studies about OSSECOs?

SM-RQ1.1 In which type of sources are articles mostly published? Our results have revealed that research on OSSECOs is mostly published in conference proceedings. The approximate ratio of publication in journals with respect to conferences is 1 to 2. This indicates that OSSECOs are considered to be a valuable software engineering research topic.

SM-RQ1.2 How has the number of publications evolved over the years? OSSECOs have been an increasingly addressed research topic since 2006. Publication peaks occurred in 2011 and 2013. There is evidence that OSSECOs have become an established research domain.

SM-RQ1.3 How are papers geographically distributed? The results in this study suggest that the current output of OSSECO papers is strongly supported by European and North American researchers. However, in the last four years, authors from other continents have been contributing with publications related to the

OSSECO topic. This review shows that the United States and The Netherlands are currently the leading countries in terms of undertaking OSSECOs.

SM-RQ1.4 Who are the predominant researchers? We observed that six authors have been the predominant researchers in OSSECOs. These authors and their clusters account for a considerable fraction of all papers covered in this systematic mapping.

SM-RQ1.5 How are publications distributed between academy and industry? It is no surprise that the publications written only by academic authors by far our number papers that have at least one industry author.

SM-RQ1.6 What type of papers are published? Although there are more empirical research papers than papers from other categories (i.e., experience reports and non-empirical papers), the difference is not significant.

2.7.2 SM-RQ2. What is an OSSECO?

SM-RQ2.1 What definitions are related to the OSSECO definition? Regarding the definitions related to OSSECOs, we encountered five major concepts (i.e., ECO, BECO, DBECO, SECO, and OSSECO), and we built a genealogical tree with their evolution.

SM-RQ2.2 Are there specific definitions of OSSECO? Our results show that there are only three definitions of OSSECOs. This thesis proposes a definition of OSSECOs, integrating the different definitions related to OSSECOs: a SECO placed in a heterogeneous environment, whose boundary is a set of niche players and whose keystone player is an OSS community around a set of projects in an open-common platform.

SM-RQ2.3 What elements belong to an OSSECO? We obtained up to 64 elements belong to OSSECOs in our review. Among them, project, community, and source code are the most used. Furthermore, we sketched a taxonomy with three categories (i.e., OSS community, ecosystem network, and software platform) to classify the OSSECOs terms.

SM-RQ2.4 What instances of OSSECOs have been reported in the literature? We identified 27 instances of OSSECOs that appear in our systematic mapping. Among them, Eclipse and GNOME are the most frequently used. we encountered five major concepts (i.e., ECO, BECO, DBECO, SECO, and OSSECO), and we built a genealogical tree with their evolution.

2.7.3 SM-RQ3. Which representations have been proposed for OSSECOs?

SM-RQ3.1 Which primary studies use models to represent OSSECOs? Our study showed that most of the papers adapt available modelling techniques or use ad hoc models to support their works, without proposing new modelling techniques.

SM-RQ3.2 Which of the proposed models, if any, are specific for OSSECOs? None of the primary studies developed a new technique, notation, or guidelines for modelling OSSECOs.

SM-RQ3.3 Which notation and guidelines have been used for modelling OSSECOs? We found a lack of specific modelling techniques for OSSECOs. However, we identified several modelling techniques to describe them in general. The most commonly applied notations were: ad hoc, tabular, and conceptual maps. Other OSSECOs were modelled using class diagrams, metamodels, or mathematical models.

SM-RQ3.4 What type of analysis was conducted using the models identified in RQ3.3? We found that most of the papers using models for OSSECOs do not conduct any OSSECO analysis. In addition, the analysis techniques used in the remaining papers, such as mathematical, visual, statistical, and SNA were used to analyze specific cases.

> *The Results of this study were published in the SCI-indexed journal Information and Software Technology (I.F.: 2.694, JCR 2016)* [34].

3 QuESo Model

"Facts are the enemy of truth."
— Miguel de Cervantes Saavedra, **Don Quixote de la Mancha**

Assessing the quality of OSS ecosystems is of vital importance because quality assurance is a way to prevent bad decisions, avoid problems, and it allows to verify the compliance with the requirements and the business goals. It can also be used for quality systematic monitoring to provide feedback and execute preventive actions. For example, before deciding to integrate a project into an established OSS ecosystem it is crucial to perform a good quality assessment to avoid problems such as inactive user communities, low level of community cohesion, or even synergetic evolution problems, i.e., lack of collaboration between the key developers.

In this chapter, we present QuESo-model for the quality assessment of OSSECOs. The chapter is divided in two complementary parts. The first describes the construction strategy of the QuESo-model V1.0. that consisted in searching the available measures for OSS ecosystems in the literature and then organize them into several quality characteristics. The second part covers an improved version of the model, named QuESo-model V2.0. This version tackled the issues presented QuESo-model V1.0 (see Section 3.2.4).

3.1 Related Work

When talking about quality models in the software domain it is inevitable to mention the ISO quality model [113]. This quality model targets the quality of a software product, from three perspectives: internal, external, and quality of use. The specific quality characteristics of the ISO quality model do not cover the important dimensions of OSS ecosystems such as the ones related to the community or the ones related to the health of the ecosystem.

The QualOSS quality model [45] gives a good representation for one of the three dimensions covered by QuESo, the OSS community. However, we had to extend it with new characteristics that are relevant in the context of OSS ecosystems (see Section 4.2 for details).

As we will explain in Section 3.2, we have found many papers that, although do not provide a quality model, they propose a good set of measures to evaluate some aspects of OSS ecosystems. We would like to mention some of these works, in particular, the ones that provided the most interesting measures.

- Hartigh et al. [114] developed a concrete measure tool for evaluating business ecosystems based on the classification made by Iansity and Lieven [115]. They conceptualized the business ecosystem health as financial health and network health based on a set of eight measures.

- Mens and Goeminne [12] provided a set of measures (e.g., number of commits, total bugs mailing list), by exploring social software engineering, studying the developer community, including the way developers work, cooperate, communicate and share information.

- Neu et al. [116] presented a web application for ecosystem analysis by means of interactive visualizations. The authors used the application to analysis the GNOME ecosystem study case.

- Kilamo et al. [117] studied the problem of building open source communities for industrial software that was originally developed as closed source.

There are different contexts in which the quality of an OSSECO is important. Adopters want to choose the right software product among similar OSS solutions in order to reduce the adoption risks. External software developers want to be motivated and interested in contributing to the OSSECO projects. Software quality models have proven useful in practice for supporting quality evaluation and assurance processes in the software domain [118], [119]. However, similar to other OSSECO abstract factors, like health [62] or resilience [108], there is a little literature available about OSSECO quality models.

Adewumi et al., [40], [120] conducted a review of the literature on OSS QMs from 2003 to 2015. They identified a total of eight OSS quality models (e.g., QSOS, OpenBRR, QualOSS, EFFORT) none of them is related to quality of OSSECOs. Also, Adewumi et al. concluded that the majority of the OSS QMs that exist today are derived from the ISO 9126 quality model and some of them (i.e., OSMM and QualOSS) satisfying all eight quality characteristics under product quality factor and usability under the quality in use factor.

On the other hand, the concepts of OSS project, OSS community and OSSECO are used interchangeable [62]. In fact, the OSS quality models are mostly quality-product oriented. In addition, some of the OSS quality include quality characteristics unique to OSS, such as: community trustworthiness, attractiveness in EFFORT [48], and activeness, heterogeneity in QualOSS [45].

3.2 QuESo-model V1.0

3.2.1 Methodology

In this section, we explain the two methodologies followed in order to design the QuESo-model. The first one is related to the way we gathered the measures from the available literature using a Systematic Mapping (SM) while the second one is related to the way we designed the QuESo-model.

3.2.1.1 Systematic Mapping (SM)

The SM protocol was described in Chapter 2, it was conducting with the goal of identifying the primary studies related to OSSECOs. The research question that addresses the measures and indicators related to the ecosystem quality is:

> *What measures or attributes are defined to assess or evaluate open source software ecosystems?*

As a result of the SM, 53 primary studies were selected, from them we identified 17 related to the identification of measures to evaluate the quality of OSS ecosystems.

Once we had collected the measures from the selected papers, we used the following criteria from [114] and [116] to include them in QuESo-model:

- User-friendly and operationalizable: measures should be logical, easy to use and operationalizable into a measurable entity.
- Non-redundant: when we identified similar measures, we selected only one of them, but we kept all the sources for traceability.

After excluding non-operationalizable and merging the similar measures with the previous criteria, we finally selected 68 different measures for the QuESo-model (note that some of the measures are used to assess more than one characteristic of the quality model).

3.2.1.2 Quality model construction

There exist several proposals for quality model construction that focus on software quality. Most of them follow top-down strategies [121], [122]. In short, they take as a basis a reference quality model such as the ISO quality model [113], take their quality characteristics as departing point and re fine them till they end up with a hierarchy with specific measures at its lower level. Remarkably, the proposal in Radulovic and Garcia-Castro [123] is mainly bottom-up oriented, i.e., it takes a set of measures as departing point to build the model. For our purposes, a bottom-up approach is the most adequate because: (1) a well-established reference quality model (or even, in its defect, a complete and systematic body of knowledge) for software ecosystems is still missing [124], and (2) there already exist a myriad of specific measures that can be applied to OSS ecosystems and that have been identified in our SM. Furthermore, although it focuses on the construction of software quality models, we can easily use it to the construction of a quality model for OSSECOs.

Radulovic and Garcia-Castro [123] proposal consists of a clearly defined sequence of steps:

1. To identify basic measures.
2. To identify derived measures.
3. To identify quality measures (by aggregation of basic and derived measures).
4. To identify relationships between measures.
5. To specify relationships between measures.
6. To define domain-specific quality subcharacteristics.
7. To align quality sub-characteristics with a quality model.

Note that the alignment in the seventh step partly implies top-down reasoning. Quality subcharacteristics that have been previously defined are related to others already specified in the existing model. If needed, some new quality subcharacteristics can be specified, or existing ones can be modified or excluded.

We have followed all the steps of the proposal. In particular, for steps 1 and 2, devoted to identify measures, we have based our work on the SM. The application of step 7 requires the use of a reference quality model. Since, to our knowledge, a quality model for the whole scope of OSSECOs is still missing, we have decided to use QualOSS [45] which measures the performance of open source communities. Clearly, new quality sub-characteristics emerging from measures related to the ecosystem considered as a whole will have to be specified, since they are not addressed by QualOSS.

3.2.2 QuESo-model

In this section, we describe the QuESo-model V1.0 obtained as a result of the application of the procedure described in Section 3.2.1. The model is composed of two types of interrelated elements: quality characteristics and measures. Quality characteristics correspond to the attributes of an open source software ecosystem that are considered relevant for evaluation. The quality characteristics are organized in a hierarchy of levels that is described in the rest of this section. The whole set of measures with their definitions is available in [125]. Also, note that we opted to keep the measure names that appear in the primary studies, even that in some cases the name given is not the most appropriate, we discuss about this topic in Section 3.2.4.

The quality characteristics in QuESo-model have been organized in three dimensions: (1) those that relate to the platform around which the ecosystem is built, (2) those that relate to the community (or set of communities) of the ecosystem and (3) those that are related to the ecosystem as a network of interrelated elements, such as projects or companies (see Figure 24. QuESo-model Figure 24).

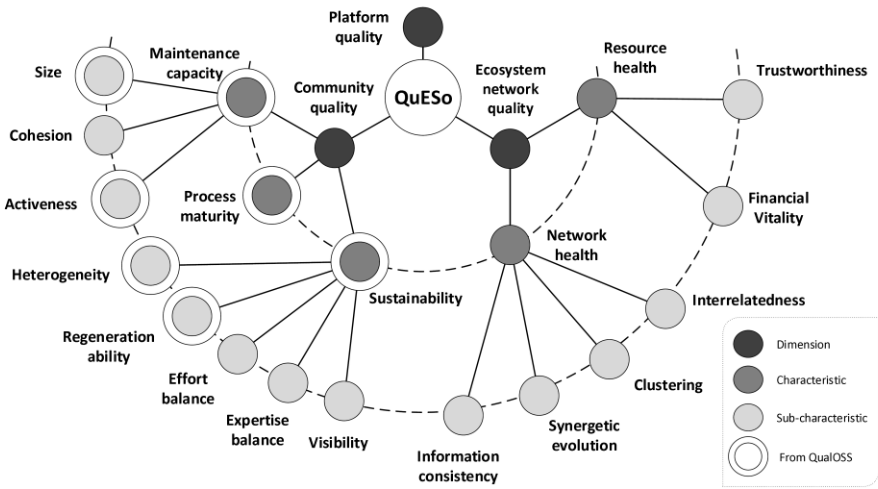

Figure 24. QuESo-model

3.2.2.1 Platform-related quality characteristics

Platform-related quality characteristics consist of the set of attributes that are relevant for the evaluation of the software platform.

As a result of our SM, we have observed that the selected papers do not provide measures for evaluating open source platform-related quality characteristics. This fact may indicate that there are not significant differential issues for open source software quality with respect to generic software quality that motivates the need of specific measures.

Then, similarly as done in the QualOSS model, since a mature proposal such as ISO 25000-SQuaRE [113] focuses on generic software quality, QuESo-model adopts directly the characteristics and sub-characteristics proposed by ISO 25000-SQuaRE and this part of the quality model is omitted in this document.

3.2.2.2 Community-related quality characteristics

Following the procedure described in Section 3.2.1, the QuESo-model proposal for community-related quality characteristics is based both on the set of measures identified in our SM and on the QualOSS quality model [45] (see Figure 24). QualOSS specifies three community characteristics, namely, *maintenance capacity, sustainability* and *process maturity*.

Maintenance capacity

Soto et al. define maintenance capacity as the ability of a community to provide the resources necessary for maintaining its products and mention that aspects relevant to it are the number of contributors to a project and the amount of time they contribute to the development effort. In order to align *maintenance capacity* with our identified measures it is refined in three subcharacteristics: *size, cohesion* and *activeness*. The *size of the community* influences its *maintenance capacity* and can be evaluated by measures such as *number of core developers* and *number of committers*. The ability of the community to collaborate defined by its *cohesion* is also relevant. A measure that can be used to evaluate *cohesion* is the ecosystem *connectedness* in the community social network. Finally, the *activeness* of the community can be evaluated by measures such as *bug tracking* activity and *number of commits*. We have identified 26 measures that can be used to measure the *maintenance capacity* (see Table 26).

Sustainability

Sustainability is the likelihood that a community remains able to maintain the products it develops over an extended period of time. According to Soto et al. it is affected by *heterogeneity* and *regeneration ability* and, as a result of our measure analysis, we have specified additional sub-characteristics besides them: *effort balance, expertise balance* and *visibility*.

The heterogeneity of a community contributes to its sustainability. For instance, if a community is mainly composed of employees of a particular company, there is the risk of the company cutting its financial support. *Heterogeneity* can be evaluated by measures such as geographical distribution of community members.

Regeneration ability also enhances *sustainability* since a community that has been able to grow in the past increases its chances of not declining in the future. A measure that we have identified for it is for instance, *new members* which counts the *number of new members of the community* at any point of time.

The *effort balance* is relevant for *sustainability* i.e., if most of the contribution effort comes from one or a small number of members of the community and it is not uniformly distributed, then its continuity is highly dependent on that small set of members. On the other hand, a balanced *effort distribution* among all members facilitates its continuity over time. Some measures for *effort balance* are: *number of developer projects* and *number of developer releases*.

Table 26. List of measures for maintenance capacity

Subcharacteristic	Measure
Size	Number contributors
Size	Number of members
Size	Number authors
Size	Number bug fixer
Size	Number of committers
Size	Number of core developers
Size	Number of nodes and edges
Cohesion	Betweenness centrality
Cohesion	Cluster of collaborating developers
Cohesion	Ecosystem connectedness
Cohesion	Out degree of keystone actors
Activeness	Bug tracking activity
Activeness	Buildup of assets
Activeness	Community effort
Activeness	Date of last commit
Activeness	Files changed
Activeness	Files per version
Activeness	Lines added
Activeness	Lines changed
Activeness	Lines removed
Activeness	Mailing list
Activeness	Number of commits
Activeness	Contributor commit rate
Activeness	Developer activity diagrams
Activeness	Temporal community effort
Activeness	Number of event people

In a similar way, the *expertise balance* among most members of a community is again a way to guarantee its *sustainability*. A community highly dependent on the expertise of one or a few members suffers from a risky situation. A measure for this is, for instance, *expertise view contributor* which calculates a contributor expertise based on the number and type of files he changed within a month.

The *visibility* of a community gives it the capacity of attracting people to contribute and support it if needed. Examples of measures identified for visibility are: *number of downloads, social media hits and web page requests.*

We have identified 28 measures that can be used to measure the sustainability quality (see Table 27).

Table 27. List of measures for sustainability.

Subcharacteristic	Measure
Heterogeneity	Geographical distribution
Reg. ability	Temporal community effort
Reg. ability	New members
Effort bal.	Contributor commit rate
Effort bal.	Developer activity diagrams
Effort bal.	Maximum number of commits of a developer
Effort bal.	Member effort
Effort bal.	Member activity rate
Effort bal.	Number of activity communities
Effort bal.	Number of developer releases
Effort bal.	Number of developer projects
Effort bal.	Project developer experience
Effort bal.	Temporal community effort
Effort bal.	Total effort of members
Exper. bal.	Expertise view contributor
Exper. bal.	Principal member activity
Exper. bal.	Relation between categorical event and developer participation
Visibility	Number of event people
Visibility	Inquires or feature requests
Visibility	Job advertisements
Visibility	Number of downloads
Visibility	Number of mailing list users
Visibility	Number of passive user
Visibility	Number of reader
Visibility	Number of scientific publications
Visibility	Social media hits
Visibility	Visibility
Visibility	Web page requests

Process maturity

Process maturity is the ability of a developer community to consistently achieve development-related goals by following established processes. It can be assessed for specific software development tasks with the answers of questions such as: is there a documented process for the task? [45]. Apparently, this characteristic requires qualitative assessment more than quantitative measures. This is consistent with the results of our SM since we have not identified measures devoted to evaluate process maturity aspects. The absence of measures for process maturity hampers the application of the bottom-up process to further refine this characteristic.

3.2.2.3 Ecosystem network quality characteristics

Since QualOSS does not address the network-related quality, this part of QuESo is exclusively based on the analysis of measures identified in our SM. QuESo proposes two ecosystem network-related characteristics: *resource health* and *network health*.

In this thesis, we take as definition for *health* applied to software ecosystems: "*longevity and a propensity for growth*" [62], [87].

Resource health

Resource health facilitates the combination of value activities from multiple actors to obtain value creating end products [126]. It is related to the financial health concept defined by Hartigh et al. [114]: *"The financial health is a long-term financially based reflection of a partner's strength of management and of its competences to exploit opportunities that arise within the ecosystem and is directly related to the capability of an ecosystem to face and survive disruptions".* In the OSS ecosystem case, this means that there is a set of partners or actors functioning as a unit and interacting among them. Their relationships are frequently operated through the exchange of information and resources. Two sub-characteristics, particularly relevant to resource health, are the *financial vitality* and the *trustworthiness* of the ecosystem.

The *financial vitality* is the viability and the ability to expand (i.e., robustness, ability to increase size and strength) of the ecosystem [127]. Two examples of measures that evaluate it are *liquidity* and *solvency* financial measures. They can be obtained directly, e.g., using balance sheet data of partners, but also indirectly, through the network relations.

Trustworthiness is the ability to establish a trusted partnership of shared responsibility in building an overall open source ecosystem [15]. Operational financial measures obtained from bankruptcy models (e.g., *Z-score* and *Zeta model*) are adequate to measure it because they take short-term and long-term survival into account [114]. We have identified 5 measures that can be used to measure the resource health quality (see Table 28).

Table 28. List of measures for resource health

Subcharacteristic	Measure
Trustworthiness	Zeta model
Trustworthiness	Z-score
Financial vitality	Liquidity
Financial vitality	Solvency
Financial vitality	Network resources

3.2.2.4 Network health

Hartigh et al. [114] define *network health* as a representation of how well partners are connected in the ecosystem and the impact that each partner has in its local network. Healthy ecosystems show many relations and subsystems of different types of elements that are intensely related [128]. Furthermore, in a healthy OSS ecosystem network, these relations are mutualistic [129]. Van der Linden et al., [130] proposed to evaluate the network health of an OSS ecosystem before its adoption. To align network health with the identified measures we have refined it into four sub-characteristics: *interrelatedness, clustering, synergetic evolution* and *information consistency*.

Interrelatedness is the ability of the nodes of an OSS ecosystem to establish connections between them. It can be evaluated by measures such as *centrality* i.e., the number of network relations of a node, and project *activity diagrams* that allows to obtain the kind of project evolution.

Clustering is the capacity of the species (or nodes) in the entire ecosystem to be classified around its projects. It also enables small OSS projects to come together as a large social network with a critical mass [131]. Basic measures as *number community projects*, *number of files* and *variety in products* can be used to identify clusters using social network analysis techniques [132].

Synergetic evolution is the ability of the subsystems that constitute the whole ecosystem to form a dynamic and stable space-time structure [127], [133]. Measures such as *ecosystem entropy* and *ecosystem reciprocity* can be used to evaluate *synergetic evolution*. The *ecosystem entropy* measure is based on the definition of software *system entropy* from Jacobson [134] who states that it is a measure for the disorder that always increases when a system is modified. Ecosystem reciprocity measures direct and active collaboration between the company and its customers in creating value propositions (e.g., through collaboration with key developers in an OSS community and other companies within the ecosystem) [135].

Information consistency is the consistency of the core information elements across the breadth of an ecosystem. The *code vocabulary map* measure evaluates this sub-characteristic. It consists of a summary of terms used in the source code of the project that can be used to obtain a general overview of the domain language of the project's network.

We have identified 15 measures that can be used to measure the *network health* quality (see Table 29).

Table 29. List of measures for network health.

Subcharacteristic	Measure
Interrelatedness	Contributor activity graph
Interrelatedness	Project activity diagrams
Interrelatedness	Networks node connection
Interrelatedness	Ecosystem connectedness
Interrelatedness	Ecosystem cohesion
Interrelatedness	Centrality
Interrelatedness	Variety of partners
Clustering	Variety in products
Clustering	Number community projects
Clustering	Number active projects
Clustering	Number of files
Synergetic evo.	Distribution over the species
Synergetic evo.	Ecosystem entropy
Synergetic evo.	Ecosystem reciprocity
Information consistency	Code vocabulary map

3.2.3 Examples of measures

In this section, we provide several examples extracted from the papers selected in the SM. In particular, we have selected the examples that belong to the Gnome software ecosystem. Our intention is to clarify the type of measures that are mentioned in this chapter with examples and also to provide some evidence of the feasibility to obtain these measures. As mentioned in [62], one of the most habitual problems is the *absence of data* to calculate the measures.

It is worth mentioning that to perform a complete quality assessment of a software ecosystem we first would need to define the assessment process which is out of the scope of this chapter (see **Chapter 5**). The quality assessment process will have to deal with, e.g., How are the values of each measure interpreted (i.e., defining what are the good and the bad values)? How can the measures be merged to provide the assessment for a particular sub-characteristic of the quality model? or What are the principles to perform the assessment with missing, incorrect, and/or inconsistent measure data? We are will provide the answer to these and other questions as part of our future work in this topic.

In the following we present the selected Gnome examples of measure values organized by the characteristics of the QuESo quality model. We omit *process maturity* because we have not found quantitative measures to evaluate it (see explanation in Section 3.2.2.2). We also omit *resource health* measures because examples for them are not reported in the SM papers for the Gnome ecosystem.

- The *maintenance capacity* can be evaluated from the *number authors* measure which gives the amount of people that change files in a project. According to Goeminne and Mens [136] data, for the Gnome ecosystem there have been 3.500 different people having contributed at least once to at least one of the Gnome projects between 1997 and 2012. The *number of commits* measure is also relevant. Each commit corresponds to the action of making a set of changes permanent. According to Jergensen and Sarma [137] approximately 480.000 commits were made in Gnome from 1997 to 2007.
- A measure for *sustainability* is the *member activity rate* which gives a value between 0 and 1 that helps to analyze the *effort balance*, i.e., a zero value indicates a uniform distribution of the work, which means that each person has the same activity rate while a value of 1 means that a single person carries out all the work. The *member activity* rate for the Gnome Evince project has had a value between 0,7 and 0,8 from 1999 to 2009 according to [16]
- The *network health* of an ecosystem can be evaluated by measures such as *number community projects* and *number active projects*. For the Gnome ecosystem, there were more than 1.300 projects between 1997 and 2012 and more than 25% of them had been active for more than six and a half years. At the lower side of the spectrum, more than 25% of all projects had been active less than one year [12]. Another measure for network health is the *contributor activity graph*. According to Neu et al. [116] one of the contributors of the Gnome ecosystem has been working in 499 projects and has more than 15.000 changes between 1998 and 2011.

3.2.4 Discussion

Some observations were made during the design of this quality model. In the following, the most interesting ones are discussed:

- *Completeness:* since we followed a mainly bottom-up strategy, the completeness of the quality model depends on how complete the set of measures found in the literature is. In this sense, we would like to remark that our quality model may be not complete by one or more of the following reasons: there may be some papers with relevant measures not included in the SM because they were not present in digital libraries or because our search string did not find them; another reason could be that some important measures are not yet reported in the literature. In this work, our intention was not to invent new measures but to organize the existing ones into a quality model.
- *Quantitative vs. qualitative:* the measures found in the literature are mostly quantitative, but a quality assessment may also include qualitative evaluations. For example, we commented in Section 3.2.2.2 the lack of measures for process maturity because in this case the assessment needs to

be done with qualitative evaluations of the community. Since we have focused on quantitative measures, there may be other characteristics of the quality model that require or that may be complemented with qualitative evaluations.

- *Unbalanced distribution of measures:* just by looking into the measure tables, it is easy to observe that the amount of measures for some characteristics is high (e.g., *activeness* with 17 measures, *visibility* with 11 measures) while for other is very low (e.g., *heterogeneity* with 1 *measure*, *information consistency* with 1 measure). This unbalanced situation could be an indicator that more research is needed for the characteristics with a low amount of measures.
- *Measure names*: we have named the measures included in the QuESo- model with the same names they are referred to in the SM papers from where they were extracted. The reason for this is to improve traceability. However, some of those measure names might be ambiguous or misleading because it is not evident from them how the measure is evaluated (e.g., *project activity diagrams*).
- *Assessment Process:* It is worth mentioning that to perform a complete quality assessment of a software ecosystem we first would need to define the assessment process which is out of the scope of this chapter (see **Chapter 4**). The quality assessment process will have to deal with, e.g.:
  - How are the values of each measure interpreted (i.e., defining what are the good and the bad values)?
  - How can the measures be merged to provide the assessment for a particular subcharacteristic of the quality model?
  - What are the principles to perform the assessment with missing, incorrect, and/or inconsistent measure data?

3.3 Updating the QuESo Quality Model

The first version of the QuESo model was described in previous section. This model allows the quality evaluation of OSSECOs. QuESo-model V1.0 defines a hierarchical structure of measures, quality subcharacteristics and quality characteristics that are organized in three dimensions: (i) those that are related to the platform around which the ecosystem is built, (ii) those that are related to the community (or set of communities) of the ecosystem and (iii) those that are related to the ecosystem as a network of elements, such as projects or companies.

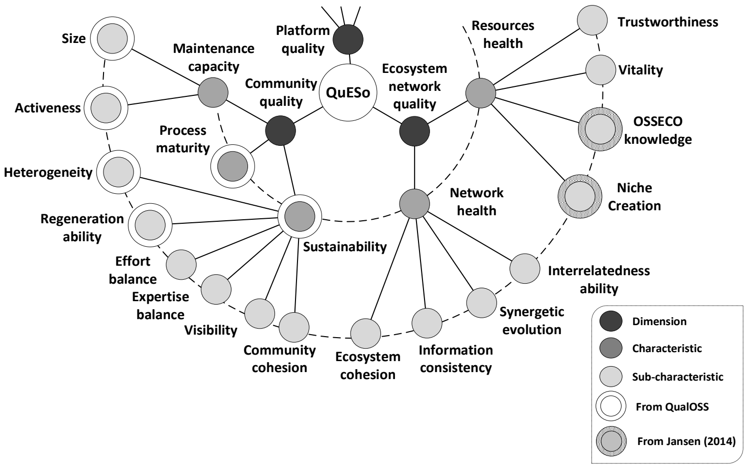

Figure 25. QuESo model

In this section, we propose a second version of the QuESo model (named QuESo model V2.0). From now in this document, we use QuESo model to refer to its current version. Figure 25 depicts a general view of the QuESo model.

3.3.1 QuESo-model V2.0 improvements

This new version improves the model in several aspects:

- Subcharacteristics improvement: In resources health quality characteristic, we added two subcharacteristics: niche creation and OSSECO knowledge. These improve the capacity of the model for evaluating the ability of OSSECOs to increase meaningful member diversity and knowledge generation over time. Furthermore, we divided the cohesion subcharacteristic into: community cohesion and network cohesion. Because, although both are related, the second one is more about a holistic point of view of OSSECOs. Finally, we extend the concept of Vitality related to metrics based on the number of distinct OSSECO members, resources and activities in time intervals.
- Measure definition improvement: We introduced a basic QuESo-measure taxonomy in order to provide a basis for a common OSSECO measure description.
- Set of measures improvement: We improved the QuESo set of measures according to the observations made in previous section: (i) completeness, we increased the number of measures from 68 to 90, (ii) unbalanced distribution of measures, the standard deviation of the number of measures per subcharacteristic was 4.3 in QuESo-model V1.0 and it is 3.6 in the current version, finally (iii) measure names, we changed the name of several measures in order to improve their understandability.

3.3.2 QuESo-model Measure Ontology

In order to describe the QuESo-model measures in detail, we introduce a basic QuESo-measure ontology. shows the graphical representation of classes and properties of the ontology. We are using the visual notation for OWL ontologies language (VOWL) [138].

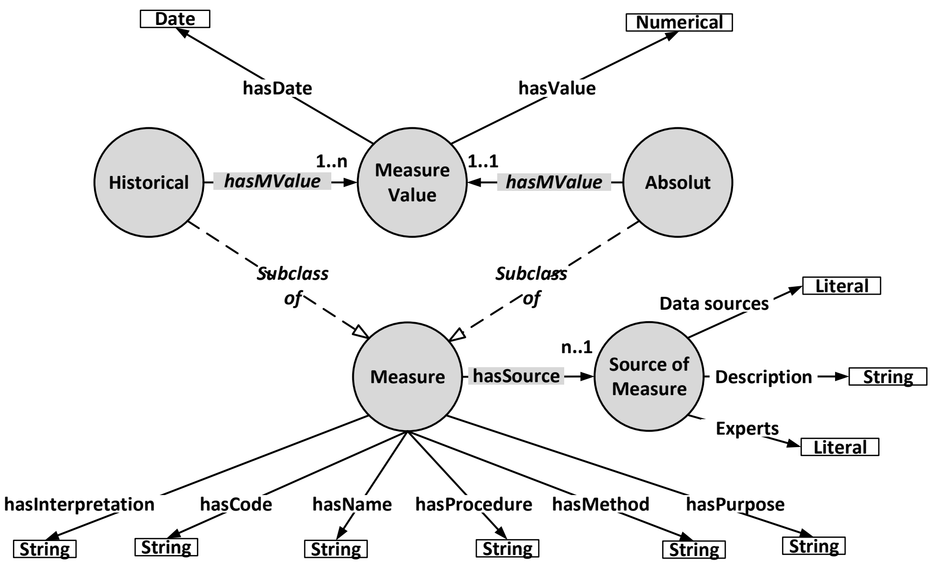

Figure 26. QuESo-model Measure ontology

3.3.2.1 Measure structure

Based on the ontology, the most important concepts in measure-definition are:

Measure

This class represents all measures in the QuESo Quality model. The list of datatype properties of this class is:

- Code: Unique identifier for the measure.
- Name: Name of the measure.
- Purpose: It is a question answered by the measure use. What does this metric tell us about the OSSECO?
- Method: Provides a summary of the application
- Procedure: Describe the measurement process or formula when is available.
- Interpretation: Provides the range and preferred values.

Source of Measure

This class specify the data repository used as a source for the OSSECO measures.

- Data sources: When the measure data can be accessed automatically using software tools.
- Experts: When it is necessary to recollect the measure data from heterogeneous experts (e.g., company adopters, OSS-community members, researchers).
- Description: A brief description of the measure data source.

Measure Value

It is a tuple with two fields: value of the measure and data when this was measured.

Historical

In this measure, the measurement process is done in a time interval. The result is a set of measure values.

Absolute

In this measure, the measurement is done in a specific time. The result is a single measure values.

A detailed list of all QuESo-model quality subcharacteristics and measures is published in [125].

3.3.2.2 QuESo-model List of Measures

In this section, we present an updated descriptions of the QuESo-model quality subcharacteristics and their related measures. These updated descriptions are made in light of the QuESo-model improvements described above (see Section 3.3.1).

3.3.2.3 Maintenance capacity

OSSECOs need a continuous input of energy in the form of new development or maintenance of the ecosystem. It is the capacity of a OSSECO to provide the resources necessary for maintaining its products [45]. In OSSECOs the maintenance capacity balances the OSS community practices and the needs of the other members of the OSSECO. In general, the objective of the *Maintenance capacity* is allowing the modification of the existing OSSECO resources, relationships and products while preserving its integrity.

Size

This subcharacteristic indicates the overall evolution of the size population in an OSSECO community. By calculating these measures for different time windows, it is possible to study the community dynamics [128]. *Size* of the community in influences, its *Maintenance capacity* can be evaluated by measures such as *Number of core developers* and *Number of committers*, these measures are typically used to compare OSSECOs [62]. Table 30 shows the *Size* measures.

Activeness

Activeness as a QuESo-model quality subcharacteristic refers to the different activity types of the OSSECO community. The community measures data come from the analysis of mailing list, forum, and bug tracking system archives, as well as from versioning system logs [45]. Table 31 shows the *Activeness* measures.

Table 30 Size Measures

Name	Purpose	Method	Interpretation	Source
Number of partners	Are there organizations that can provide different types of support to an OSSECO-community?	Count the number of companies, institutions, research communities, etc. that support the OSSECO-community.	More is better. More partnerships indicate a strong project that is well embedded in the community.	OSSECO websites.
Number of passive users	How many people are just downloading and using the software produced by the OSSECO-Community?	Count the number of OSSECO-community members from mailing list that are not contributors.	More is better. Passive users are essential for a sustainable OSSECO-community.	OSSECO repositories.
Number of contributors	How many people are collaborating in different types of activities in the OSSECO-community?	Count the number of people that make changes in the OSSECO-community data repositories e.g. authors, active users, committers, readers, translators.	More is better. Number of active and mature contributors is a measure that indicates a healthy OSS-community.	OSSECO repositories.
Size of network community	What is the size of the OSSECO-community social network?	Count the number of edges in the network. OSSECO-community social network its a graph were nodes are members and the edges are different type of relationships (e.g., emails, shared commits).	More is better. Networks with more nodes and connections are bigger and have a better structure.	OSSECO repositories.
Number of OSSECO-community members	How many people are in the OSSECO-community?	Count the number of contributors, passive users and partners in the OSS-community.	More is better. More members indicate that the OSS-community have a good structure for maintaining its products.	OSSECO repositories.

Table 31 Activeness Measures

Name	Purpose	Method	Interpretation	Source
Bug tracking activity	Is the OSSECO-community active in the bug tracking system?	Count the number of bugs activities (e.g., bugs opened, closed, status changed, bug fix time, etc.) from the project bugs tracker system.	Bug tracking activity is an indicator of OSSECO-community activity.	OSSECO bug-tracking system.
Communication and bug fixing correlation	Is there a correlation between OSSECO-developers communication and software quality?	Calculate the number of developers communication in mailing list and the number of bugs fixed in a OSSECO-project.	Perfect negative correlation is better.\newline There is a significant correlation between communication and quality	OSSECO mailing lists and bug tracking system.
Date of last commit	How is the OSS-community actuality?	Read the date of the last commit from the version control repository system.	More recently is better\newline A OSS-community that has more actuality is more active.	OSSECO version control repository system.
Version history	How is the OSS-community evolving?	Calculate the number of major releases per year from the OSS-community web	More is better. The number of project releases show the evolution in an OSSECO-community.	OSSECO web sites.
Date of last release	How is the OSSECO-community actuality?	Read the date of the last release from the version control repository system.	More recently is better\newline A OSS-community that has more actuality is more active.	OSSECO version control repository system.
Number of changed files	How is the activity in the OSSECO-community repositories?	Count the number of files changed per commit.	More is better\newline High number of files changed implies more OSS-community activity and effort .	OSSECO version control repository system.
Number of files per version	How is the size evolution of the OSS-community projects?	Count the number of files of all projects belonging OSSECO-community from the version control repository system.	More is better. An activity OSS-community create many files.	OSSECO version control repository system.
Communication activity	How is the communication between OSSECO-community members?	Read the OSSECO-community communications activity from the e-mail system.	More is better\newline High communications activities implies a high active OSSECO-community.	OSSECO mailing lists.
Community contributor activity	How is the commits activity in the OSSECO-community projects?	Count the number of commits and count the number of contributors from the control repository system.	High is better\newline Commits activity is an indicator of OSS-community activity.	OSSECO version control repository system.
Community commit rate	How long between commits of a OSS-community?	Calculate the average of time between first and last commits dates of all OSSECO-community projects.	Low is better\newline Smaller rate can be interpreted like a more active OSSECO-community.	Version control repository system.
Decline point	When the values of number of emails starts to decline?	Calculate the date for which the number of contributors is an 80\% of the culminating point.		OSSECO Mailing lists.
Culminating point	When the number of emails is maximum in the OSS-community history?	Calculate the date for which the maximum number of emails is found in the OSSECO-community history		OSSECO Mailing lists
OSSECO-community activity period	Is it possible to know whether a OSS-community was active all the time or they have been some gaps of inactivity during its lifetime?	Identify the period of time between culminating point and decline point	Large is better\newline A large community activity period is better	OSSECO Mailing lists.
Community Timelines	How is the OSSECO-community timeliness?	Calculate the average of time between date of request and date of first response of all projects in the OSSECO-community	Low is better.	OSSECO mailing lists. version control repository system and bug tracking system.
Number of events	How is the social activity in a OSSECO-community?	Count the number of events and people that participate in these events	More people are better and more events are better.	Content management system

3.3.2.4 Sustainability

According to [139] *Sustainability* is one of the fundamentals challenges in any type of ecosystem. A sustainable natural ecosystem maintains its characteristic diversity of major functional groups, productivity, and rates of bio-geochemical cycling, even in the face of disturbing events. Similar to Dhungana et al. [139], we defined a sustainable OSSECO to be the one that can increase or maintain its products, resources, members and relationships over longer periods of time and can survive inherent changes such as new technologies, new products, competitors that can change the population. In summary *Sustainability* is the likelihood that a OSSECO remains able to maintain the products or services it develops over an extended period of time.

Heterogeneity

The *Heterogeneity* of a community contributes to its *Sustainability*. For instance, if a community is mainly composed of employees of a particular company, there is the risk of the company cutting its financial support. *Heterogeneity* can be evaluated by measures such as *Geographical distribution* of community members. Table 32 shows the *Heterogeneity* measures.

Table 32 Heterogeneity Measures

Name	Purpose	Method	Interpretation	Source
Geographical members distribution	Are the members of the OSSECO-community geographically distributed?	Identify the geographical location of members from the mailing lists. Count the number of the different geographical locations (e.g. countries).	More is better. More members distributed implies more heterogeneity.	OSSECO mailing lists
Member activity types	Are the OSSECO-community member distributed across different activity types?	identify the activity types from the file paths and filenames. Count the number of members participating in each activity.	Closer to 0 is better A zero value for these indices implies a uniform distribution. A value of 1 means inequality.	OSSECO version control repository system.
Variety in OSSECO-community projects	How many kinds of projects has the OSSECO-community?	Search project information in OSSECO-community data sources.	More is better. A large variety in projects is an indicator that there are many niches, platforms, domains, etc., in which a new player can become active.	OSSECO projects index, content management system and multi-homing .
Variety of OSSECO-community partners	How many kinds of partners has the OSSECO-community?	First partners are classified into species by their characteristics (e.g., private, public). Second calculate the proportions of the species in the entire market as a reference point Calculate for each partner the proportions of different species that is related to.	High is better. Covariance with market indicates the variety of different partners a partner has.	OSSECO-partners surveys
Organizations members' distributions	How are the affiliations of OSSECO-community members to organizations?	Count the number of organizations in which the OSSECO-community members are affiliated.	More is better.	OSSECO-partners surveys.

Regeneration ability

Regeneration ability is the degree to which the size evolution of a OSSECO community happens at an adequate rate to maintain a sustainable community size that allows them to survive the loss of some of their human resources [31]. *Regeneration ability* also enhances *Sustainability* since a community that has been able to grow in the past increases its chances of not declining in the future. A measure that we have identified for it is for instance, *New members* which counts the number of new members of the community at any point of time. Table 34 shows the *Regeneration ability measures*

Table 34 Regeneration ability Measures

Name	Purpose	Method	Interpretation	Source
Contributor survival rate	What is the number of surviving contributors in the OSSECO community?	Calculate the OSSECO community contributors that were at the beginning and survived in a period time.	Higher is better.	OSSECO mailing lists, version control repository system and bug tracking system.
Community new members rate	Is the number of OSSECO-community members evolving?	Count the number of members that have done their affiliation to the community on a time period.	Higher is better. More new members more survival likelihood.	OSSECO mailing lists.
Community new contributors rate	Is the number of OSSECO-community contributors evolving?	Count the number of contributors that had done their first contribution to the OSSECO-community on a time period.	Higher is better. More new contributors more survival likelihood.	OSSECO mailing lists, version control repository system and bug tracking system.

Effort balance

Effort balance is relevant for *Sustainability* i.e., if most of the contribution effort comes from one or a small number of members of the community and it is not uniformly distributed, then its continuity is highly dependent on that small set of members. On the other hand, a balanced effort distribution among all members facilitates its continuity over time. Some measures for effort balance are: *Number of developers* per project and *Number of developers* per release. Table 33 shows the *Effort balance* measures.

Table 33 Effort balance Measures

Name	Purpose	Method	Interpretation	Source
Contributors commit time rate	How long between commits of all contributors?	Calculate the time between commits. Calculate variance of time commits for a contributor. Calculate variance of variance of all commits contributors time.	Lower is better.	OSSECO version control repository system.
Community project involvement	How are the commits contributions distributed in the OSSECO-community projects?	Count the number of each activity in each time interval per OSS-community project.	Similar percentage of contributions is better	OSSECO version control repository system.
Statistical characteristics of commits	How is the variation of commits across OSSECO-community history?	Calculate statistical general values of commits from history data from data sources	High values are better.	OSSECO version control repository system.
Community activities rate	Are the activities distributions balanced in the OSSECO-community projects?	Compute the Gini Index for all or subset of: commits, mails sent, files changed and bug report reading the data in several time intervals in the OSSECO-community projects.	Closer to 0 is better A zero value for these indices implies a uniform distribution. A value of 1 means inequality.	OSSECO mailing lists, repository system, bug tracking system.
Sub-communities of members rate	How many sub-communities are the developer involved?	Count the number of sub-communities in which all member are involved in the OSS-community	It depends of the context	OSSECO mailing lists.
Developer release rate	How is the developers participation in the OSSECO-community?	Count the number of releases in which each developer has been active on a project.	Lower is better	OSSECO repository system.

Expertise balance

Developing complex software projects in a OSSECO community requires skill and expertise in a specific domain (e.g., Eclipse community requires knowledge and expertise in plugins, R OSSECO requires skills in statistics). Expertise is one of the most overarching attributes of OSSECO communities [137]. Adopters and contributors typically share the same level of technical expertise (i.e., mostly developer-to-developer communication [51]). The *Expertise balance* among most members of a community is again a way to guarantee its *Sustainability*. A community highly dependent on the expertise of one or a few members suffers from a risky situation. Table 35 shows the *Expertise balance* measures.

Visibility

Table 35 Expertise balance Measures

Name	Purpose	Method	Interpretation	Source
Contributors expertise	How is the expertise of the OSSECO-community contributors?	Count the number of files that contributors have changed and classify this by their file extension.	High balanced is better. A zero value for these indices implies a uniform distribution. A value of 1 means inequality.	OSSECO version control repository system.
Longevity of contributor rate	How many time is a developer in the OSSECO-community?	Obtain dates of first and final author commits. Calculate time between dates	High is better.	OSSECO version control repository system.
Contributor experience rate	How is the experience of the OSSECO-community contributors?	Count the total number of releases in which the contributors were active. Contributor project experience= Number of releases since the contributor's first activity on the project.	High is better.	OSSECO version control repository system.
Number of projects per contributor	Does each OSSECO-ecosystem contributor have a OSSECO-community project in which it contributes?	Count the number of projects where contributor	Close to 0 is better. A zero value for these indices implies a uniform distribution. A value of 1 means inequality.	OSSECO version control repository system.

Visibility of a community gives it the capacity of attracting people to contribute and support it if needed. This can be measured as an aggregation of several measures such as the *Number of events*, *Number of patents* etc. Table 36 shows the *Visibility* measures.

Table 36 Visibility Measures

Name	Purpose	Method	Interpretation	Source
Number of members making new features requests	How many members are making new features requests?	Count the number of OSSECO-community members that are making new features or inquiries	More is better.	Version control repository system. bug tracking system.
Number of Job advertisements	Are there job opportunities for the OSSECO-community members?	Search job advertisements for OSSECO-community members.	More is better.	Specialized social networks and web sites.
Number of downloads	Are the OSSECO-ecosystem projects popular?	Count the number of downloads of the OSSECO-community projects.	More is better.	OSSECO download web pages and version control repository system.
Number of OSSECO-community mailing list subscribers.	What is the contributors OSSECO-community size?	Count the number of contributors subscribed to the OSSECO-community mailing lists.	More is better.	OSSECO mailing lists.
Number of passive users	it is equal to S-NoPU			
Number of scientific publications referencing the OSSECO-community	Are the OSSECO-community data sources used by researchers?	Search for scientific publications about the OSSECO-community in scientific libraries and databases.	More is better.	Digital libraries and scientific databases.
Web and Social media hits.	Have the OSSECO-community projects visibility on the web	Count the number of hits of the project in: blogs and social media.	More is better.	Blogs and social media web sites.
Number of patents	How many patents has the community?	Count the number of patents of the OSSECO-community.	More is better.	Community members.
Number of events	How is the social activity in the OSSECO community?	Count the number of events and people that participate in these.	More people are better. More events are better.	OSSECO: content management system.
Contributor ratings and reputation.	How is the OSSECO-ecosystem contributors reputation?	Obtain contributors data from OSSECO-community data sources and surveys.	High is better. High contributors ranking implies more Trustworthiness.	OSSECO contributors OSSECO web pages.
Geographical members distribution	It is equal to H-GMD			
Community acceptance	How is the OSSECO-community acceptance by commercial organizations?	Obtain information from partnerships about OSSECO-ecosystem acceptance.	High is better. OSSECO-community acceptance is better.	Experts: OSSECO partners.
Number of web pages referencing the OSSECO-community web page	How many page referenced the OSS-community web page?	Use a specialized software for count the number of web pages.	More is better.	Data sources: World Wide Web.

Community cohesion

Cohesion is an indicator of connectedness between members in a OSSECO-community. It is a property that keeps communities' structure safe from risks, guaranteeing their wellbeing and health [140]. Cohesion guarantees an efficient exploitation of core resources, a proper ow of information between members and provides the necessary protective factors against the entrance of new competitors [114]. Table 37 shows the *Community cohesion* measures.

Table 37 Community cohesion Measures

Name	Purpose	Method	Interpretation	Source
Community betweenness centrality rate	What is the ability of a node to act as a mediator in the community?	Calculate the betweenness centrality for each node in the network, e.g., nodes can be contributors and edges are messages in the mailing list. Calculate the percentage of nodes with betweenness centrality > 0	Closer to 1 is better	OSSECO mailing lists and version control repository system.
Cluster of collaborating developers	How to identify clusters of developers in an OSS-community?	Define a social network: The nodes are developers and the edges between them represent projects on which the collaborated. Developers which collaborate are positioned closer together.	A cluster per project is better.	OSSECO mailing lists and version control repository system.
Outbound links to other OSSECO sub-communities	How well the OSSECO sub-communities are connected between them?	Identify resources dependencies and construct a network dependencies graph.	High is better. Outbound links are a measure of robustness.	OSSECO version control repository system.
Outdegree of keystone actors.	There are keystone actors in the OSS-community?	Define a social network: The nodes are members of OSSECO-community and the edges between them represent any activities on which they collaborated.	Network with keystone actors are better connected is better	OSSECO mailing lists, version control repository system, bug tracking system.

Table 38 Ecosystem cohesion Measures

Name	Purpose	Method	Interpretation	Source
Number of nodes to disconnect the OSSECO-ecosystem	How is the OSSECO-ecosystem connected?	Define a OSS-ecosystem social network. Calculate the minimum number of nodes that would need to be removed from the network before it becomes. disconnected.	More is better.	OSSECO version control repository system, mailing lists, content management system. Experts: OSSECO members.
Outdegree of keystone actors	How many keystone actors have he OSSECO?	Define a OSSECO-ecosystem social network. The outdegree of a node v is the number of edges with v as their initial vertex.	High is better. High outdegree implies that the actor plays a large role in the OSSECO.	OSSECO mailing lists, version control repository system and bug tracking system. Experts: OSSECO members.
OSSCO clustering coefficient	How close are the OSSECO nodes to becoming a complete graph with its neighbors?	Define a social network: The nodes are the OSSECO members/artifacts and edges definitions depend of the type of the ecosystem analysis. (mails, code, project dependencies, resources).	Close to 1 is better.	OSSECO mailing lists, version control repository system and bug tracking system. Experts: OSSECO members.
Number of partners connections	How many connections have the partners in the OSSECO network?	Define a network: partners and projects are the nodes and any communication is a edge. Count the number of connections between central and non-central species or partners.	More is better. More number of connections implies more Interrelatedness.	Data source: version control repository system. Experts: OSSECO partners.

3.3.2.5 Network health

Ecosystem cohesion

This subcharacteristic is related to the *Community cohesion* subcharacteristic, the difference is that it takes a holistic point of view of the OSSECO for cohesion. Table 38 shows the *Ecosystem cohesion* measures.

Information consistency

The communication in a OSSECO has a common vocabulary that presents the summary of the terms used in the OSSECO. A common vocabulary is a tool for the members who wants to obtain a general overview of the domain language of a OSSECO. Table 39 shows the *Information consistency* measures.

Table 39 Information consistency Measures

Name	Purpose	Method	Interpretation	Source
OSSECO synonyms in vocabulary map	How to obtain a general overview of the domain language of the OSSECO?	Construct a vocabulary map with the terms in the OSSECO data sources. Identify synonyms	A common language is better. Less synonyms are better	OSSECO version control repository system, mailing lists and content management system.
OSSECO sentiment analysis	How is the message vocabulary content in the OSSECO?	Pre-process the email messages. Configure the sentimental words. Score the words. Classify the words.	A positive vocabulary is better.	OSSECO mailing lists.

Synergetic evolution

It is the ability of the subsystems that constitute the whole ecosystem to form a dynamic and stable space-time structure [96], [127]. Synergetic evolution measures the collaboration between the key members in an OSSECO. Table 40 shows the Synergetic evolution measures.

Interrelatedness ability

Interrelatedness *is the ability of nodes in an OSSECO to establish connections between them based on the ways developers collaboratively contribute to the OSSECO projects [50]. Table 41 shows the* Interrelatedness ability *measures.*

Table 40 Synergetic evolution Measures

Name	Purpose	Method	Interpretation	Source
OSSECO partners distribution	Is the partner distribution over the ecosystem species equality?	Count the number of partners in each OSSECO project	Close to 0 is better. A zero value for these indices implies a uniform distribution. A value of 1 means inequality.	Experts: OSSECO partners.
OSSECO popularity	How is the OSSECO popularity in external companies?	Calculate the information entropy	High is better. If entropy is low the OSSECO is supported by a small number of companies.	Experts: OSSECO partners. OSSECO sponsors.
OSSECO projects community partnership and embeddedness	How is the OSSECO-community projects embeddedness?	Obtain information about partnership model.	High is better	OSSECO Content management system. Experts: OSSECI members
OSSECO reciprocity	Is there OSSECO reciprocity?	Define a social network: The nodes are the OSSECO members/artifacts and edges definitions depend of the type of the ecosystem analysis (e.g., mails, code, project dependencies, resources).	P =0 is better. This measures the amount of direct reciprocity (P>=0) or anti-reciprocity (P <0) in networks, with mutual links occurring more and less often than in random networks. The neutral or reciprocal case corresponds to p=0.	version control repository system, mailing lists and content management system. Experts: OSSECO members.

Table 41 Interrelatedness ability Measures

Name	Purpose	Method	Interpretation	Source
OSSECO partners connectedness evolution	How is the partners connectedness evolution in the OSSECO?	Define a OSSECO social network. Number of relations as a proportion of the theoretically maximum number of relations in all network.	Growing is better. High connectedness is a property that keeps OSSECO structure safe from risks, guaranteeing their well-being and health.	OSSECO version control repository system, mailing lists, content management system. Experts: OSSECO members.
OSSECO partners with other OSSECO members connectedness evolution.	How is the partners connectedness with other OSSECO members evolution?	Define a OSSECO social network. Number of relations as a proportion of the theoretically maximum number of relations in all network.	Growing is better. High connectedness is a property that keeps OSSECO structure safe from risks, guaranteeing their well-being and health.	Version control repository system, mailing lists and content management system. Experts: OSSECO members.
OSSECO centrality evolution.	What OSSECO members tend to be more connected between them?	Define a OSSECO social network. It's possible to calculate several centrality measures: C1= Partner centrality. C2= Project code centrality. C3= Ecosystem code centrality. C4= Files centrality. C5= Centrality of a developer's contributions.	The centrality is used within network analysis as a measure to indicate the importance of a node in the network.	OSSECO version control repository system, mailing lists and content management system. Experts: OSSECO members.

3.3.2.6 Resources health

In business ecosystems (BECOs), *Resources health* is related to the financial health concept defined by Hartigh et al. [114]: “it is a long-term financially based reflection of a partner's strength of management and of its competences to exploit opportunities that arise within the ecosystem and is directly related to the capability of an ecosystem to face and survive disruption”. In the context of OSSECOs, we defined *Resources health* as the ability of an OSSECO to obtain value from their symbiotic relationships between all members and resources of the ecosystem. This means that the OSSECO will remain growing and increasing in longevity [114].

Niche creation

It is the ability of the OSSECO to increase meaningful members diversity over time [142]. According to [143] *Niche creation* describe how much opportunity there is in the OSSECO to start as a new niche player.

Table 42 Niche creation Measures

Name	Purpose	Method	Interpretation	Source
Number of context types of OSSECO projects applications	Have the OSSECO projects different types of context applications?	Identify the project OSSECO dependencies. Crawl the OSSECO content management system. Identify the project contexts.	More is better. A wide variety of OSSECO project applications contexts, will be more supporting for niche creation.	version control repository system and content management system experts: OSSECO members.
Number of natural languages supported	Is the OSSECO multi-language?	Crawl the OSS content management system. Identify different languages in the OSSECO repository.	More is better. A wide variety of supported languages, will be more supporting for niche creation.	OSSECO content management system.
Variety in OSSECO project technologies.	Does the OSSECO projects support different technologies?	Read information from OSSECO data sources. Identify the OSSECO development technologies.	wide variety of technologies, will be more supporting for niche creation.	OSSECO version control repository system and content management system. Experts: OSSECO members.
Number of OSSECO platform extensions	How many platform extensions have the OSSECO?	Obtain the number of extensions from the version control repository system.	More is better. Each extension is a potential ecosystem niche	OSSECO content management system.
Number of niches of the OSSECO	How many niches have the OSSECO?	Obtain the number of niches.	More is better.	OSSECO version control system. Experts: OSSECO members.
Number of markets in which the OSSECO is participating	Are the OSS-ecosystem projects in multiples markets?	Identify the project dependencies.	More is better. A wide variety of markets, will be more supporting for niche creation.	Version control repository system. End-users survey. Content management system.

Furthermore, it is one of the measures defined by [115] for evaluating ecosystem health. Niche creation is also, one of the OSEHO platform pillars (Open Source Ecosystem Health Operationalization) defined by [143]. Table 42 shows the *Niche creation* measures.

SSECO Knowledge

The open and shared development practices in OSSECOs allow to contributors adding knowledge such as aggregated information, blog posts, and manuals into a common knowledge base and code repositories, indicating also that the OSSECO is healthy [62]. Table 43 shows the OSSECO *Knowledge* measures.

Table 43 OSSECO Knowledge Measures

Name	Purpose	Method	Interpretation	Source
Number of activity types	How many types of activities has the OSSECO?	Obtain the type of activity of the OSSECO members.	More is better.	OSSECO version control repository system. Experts: OSSECO members.
Number of OSSECO artifacts	Are the contributors adding knowledge to the OSSECO?	Crawl (Counting the knowledge artifacts). Some artifacts can be: A= Blog posts. B= Manuals. C= Translations. D= Marketing materials. E= Scientific papers.	More is better.	OSSECO content management system, OSSECO Wikis and digital libraries.

Vitality

Vitality is the viability and the ability of an OSSECO to expand (i.e., robustness, ability to increase size and strength) of the ecosystem [127]. *Vitality* related metrics are based on the number of distinct OSSECO members, resources and activities in time intervals. Table 44 shows the *Vitality* measures.

Trustworthiness

OSSECO *Trustworthiness* is the ability to establish a trusted partnership of shared responsibility in building an overall open source ecosystem [15]. Operational financial measures obtained from bankruptcy models (e.g., Z-score and Zeta model) are adequate to measure it because they take short-term and long-term survival into account [114]. Table 45 shows the Trustworthiness measures.

Table 44 Vitality Measures

Name	Purpose	Method	Interpretation	Source
Liquidity of partners	Can the partner to meet your short-term obligations?	Obtain the financial data from the partner. Calculate de Liquidity for each partner in the OSSECO. Count the number of partners with liquidity < 1.	Less is better. X < 1 is dangerous for the partner. 1\textless= X<= 2 is normal. X > 2 is good.	OSSECO public partners data. Experts: OSSECO members.
Market share of OSSECO projects	How is the OSSECO projects market shared?	We need to do end-user surveys in order to collect the knowledge and information that is already available, such as market reports, open source evaluations, and other platform popularity data. Finally, on an aggregate level we can analyze, using source code and manifest analysis, how frequently a project is required and used by other projects.	More is better.	OSSECO content management system, mailing lists and code repositories. Furthermore, if it is possible, to identify the OSSECO-code inclusion in other projects and software ecosystem Experts: OSSECO final-users.
Solvency of partners	Can the OSSECO partners pay their debts?	Obtain the financial data from the partner. Calculate de solvency for each partner in the OSSECO. Count the number of partners with solvency \textless 1.	Low is better. 0<= X < 1 is normal 0<=1/3 may be financial problems X< 0 negative equity. X > 1 conservatively financed \c.	OSSECO partners public data. Experts: OSSECO partners.
Buildup of assets of partners	How is the OSSECO partners productivity?	Obtain financial data from the partners. Calculate the total factor productivity over time.	High is better	OSSECO partners public data. Experts: OSSECO partners.
Limited obsolescence	Are the OSSECO infrastructure obsolete?	Obtain information about OSSECO technologies.	No obsolete is better.	OSSECO content management system. Experts: OSSECO members.
Continuity of use experience and use cases	How to the OSSECO evolve in response to new technologies?	Obtain list of projects releases technologies information. Compare releases evolve with OSSECO-community platform evolve and programing languages and operating system evolving.	Evolve more rather than changing abruptly is better.	OSSECO: Content management system. General context of the technologies.
OSSECO community acceptance	How is the OSSECO community acceptance by commercial organizations?	Obtain information from partnerships about OSSECO acceptance.	High OSSECO acceptance is better.	OSSECO Content management system Experts: OSSECO Partners.
Number of passive users	it is equal to S-NoPU		More is better. Passive users are essential for a sustainable OSS-ecosystem.	
Number of new communities	Are the OSSECO creating new communities continuously	Count the number of new communities in a period of time.	More is better.	OSSECO mailing lists and version control repository system.
OSSECO partnership and embeddedness	How is the OSSECO embeddedness?	Obtain information about partnership model.	High embedded is better.	Version control repository system.

Table 45 Trustworthiness Measures

Name	Purpose	Method	Interpretation	Source
ZETA score of partners	How is bankruptcy score of the OSSECO partners?	Obtain the financial data from the partner. Test the creditworthiness and solvency of a partner. Count the number of OSSECO partners with ZETA score below 1.8. ZETA score =1.2A+1.4B+3.3C+0.6D+1.0E A = Working Capital/Total Assets B = Retained Earnings/Total C = Earnings Before Interest Tax/Total Assets D = Market Value of Equity/Total Liabilities E = Sales/Total Assets.	High is better. A score below 1.8 means the company is probably headed for bankruptcy.	Partners public financial data. Experts OSSECO partners.
OSSECO Age	How old is the OSS-ecosystem?	Calculate lifetime of the OSSECO	Older is better.	OSSECO version control repository system.
Number of patents of partners	How many patents has the OSSECO partners?	Count the number of patents of the OSS- ecosystem partners.	More is better.	OSSECO partners.
Contributor ratings and reputation	How is the OSSECO contributors reputation?	Obtain contributors data from OSSECO data-sources and surveys.	High is better. High contributors ranking implies more Trustworthiness.	OSSECO content management system. Experts: OSSECO members.

3.4 Conclusions

In this chapter, we have presented QuESo, a quality model for assessing the quality of OSS ecosystems. This quality model has been constructed following a bottom-up strategy that consisted in searching the available measures for OSS ecosystems in the literature and then organize them into several quality characteristics.

The presented quality model covers three aspects of OSS ecosystems: the platform, the community, and the ecosystem network; which altogether are a good representation of the most important aspects of an OSS ecosystem.

This quality model can be used as a starting point for the quality assessment of an OSS ecosystem, and it is in our plans for the future work to define a complete quality assessment process (as described in Chapter 4) and to apply it in a real quality assessment. Finally, we presented an update of the QuESo-model.

The results about QuESo-model were published in [20], [125], [58]

4 QuESo Process

"Some things come too soon and others come too late, but we only find out when there's nothing to be done, when we've already bet against ourselves."
— **Alvaro Mutis**, **The Adventures and Misadventures of Maqroll**

The aim of this chapter is to describe the activities of the QuESo-process, this is one artefact of the QuESo-framework designed for the quality evaluation of OSSECOs (RQ2). We propose an approach based on building Bayesian networks (BN) as evaluation model. These networks are derived systematically from: the QuESo-model, historical data from OSSECOs and empirical survey data from experts.

4.1 Related Work

Currently, there is little literature available about process, frameworks or methods to evaluate the quality of OSSECOs. Furthermore, there exists no working operationalization available that can be used to determine the quality of OSSECOs. In a recent systematic mapping study Axelsson and Skoglund [49] investigate the challenges related to quality assurance in SECOs. However, this study only resulted in a set of six papers, none of which propose a process for quality evaluation of OSSECOS. Nonetheless, in their work, the literature findings were mapped to life-cycle process based on ISO 12207 in light of its four high-level process areas: agreement processes, organizational project-enabling processes, project processes and technical processes. Furthermore, Axelsson and Skoglund present a research agenda, wherein they claim for more research in life-cycle processes in quality assurance in software ecosystems. In the same way, in an earlier research agenda, Jansen et al., [24] mention quality management as one of several challenges in software ecosystems as well.

Amorim et al. [144] present a work in progress about a three-step method (named EMSEP) for evaluating the essential practices that are commonly used in SECOs (e.g., coordination, market share, quality management, etc.). Goeminne and Mens [28] propose a framework to analyze the evolution of OSSECOs. Furthermore, they present Herdsman, a tool for a dynamic visualization of a set of metrics that represent the OSSECO evolution.

In regards to quality model life-cycle, the work of Mens et al. [145] present an application of the Model-Centric Quality Assessment (MoCQA) framework. They instantiate this framework to measure the quality of evolving libre software distribution ecosystems. MoCQA provides a quality model life-cycle that is adequate for the notion of SECO. Finally, Fotrousy et al. [146] give an overview of existing research on Key Performance Indicators (KPIs) for the assessment of SECOs. They identified those KPIs that are used to enable quality management of SECOs, in particular: performance, usability, security, data reliability, extendibility, transparence, trustworthiness and quality-in-use KPIs.

4.2 Background

OSSECOs have emerged in the last few years as a novel way to understand the relationships between software projects, products, and organizations in the OSS context[58]. Because of this, the capability to perform a quality evaluation of this type of SECOs is becoming increasingly important for companies, institutions and stakeholders. This evaluation should provide relevant indicators, as well as *"incorporating the inevitable uncertainty, reliance on expert judgment, and incomplete information that are pervasive in software engineering"* [147].

On the other hand, there has been considerable interest to use Bayesian networks in software engineering (e.g., [148], [119], [118]) . In fact, this approach has been largely used as an efficient tool for knowledge representation and inference in law, medicine, finance and engineering [147], [149]. The approach to use BNs for assessing software quality has been developed foremost by Fenton & Neil. These allow representing quality models by modelling the relationships between software measures, quality subcharacteristics and quality characteristics in order to evaluate the quality of software artefacts [148].

In this section, we briefly introduce the definitions and basic properties of quality models and Bayesian networks.

4.2.1 Quality Models

In the ISO/IEC 25010 standard, software quality is defined as the following: "*the degree to which a software product satisfies stated and implied needs when used under specified conditions*" [150]. Quality models describe what is intended by quality for a particular purpose and domain and refines this concept in a structured way [119].

Usually the software quality characteristics (e.g., *reusability, usability, portability*) and software quality subcharacteristics (e.g., *suitability, security, understandability*) are hierarchically organized [151], [120]. The leaves of this hierarchy represent quality measures which can be directly measured and assessed. Each measure is related to one or more quality subcharacteristics. In addition, empirical relationships between measures, and between measures and quality subcharacteristics are established. These relationships must be enhanced with logic or expert assessment in order to be considered as representing cause-effect between measures and quality subcharacteristics [148]. Figure 27 depicts a generic quality model structure based on the ISO quality model [152]. Moreover, SECOs are also underrepresented in software quality models, because aspects such as open repositories, community entropy and high frequency of commits and releases are not present. Therefore, by combining these three points of view (software, OSS, and ecosystem) we produced a quality model for OSSECOs, the QuESo-model [20] (see Chapter 3 for more details).

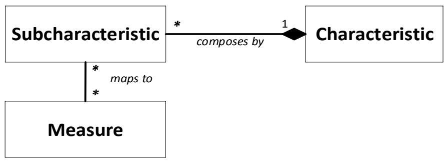

Figure 27. Figure QM structure

4.2.2 Quality Evaluation

In order to use a quality model in a practical domain, it is necessary to define an evaluation process of a quality aspect (i.e., the focus of the OSS artefact that is addressed by the quality model) [118]. For example, software products, OSS community or OSS health. This process should also define quality in a quantitative manner (i.e., concrete measures and evaluations are obtained). Furthermore, the measurement and evaluation process require a mechanism for determining what data is to be collected, why it is to be collected, and how the collected data is to be interpreted [153], [154]. In conclusion, quality models should be associated with a systematic method in order to evaluate the software quality [155].

4.2.3 Bayesian Networks

Causal probabilistic networks, known as BNs, are essentially a directed acyclic graph (DAG) for representing conditional probabilistic distribution (CPD) over a set of nodes or variables. This CPD is often calculated from a node probability table (NPT) [156]. As stated by [119], "*these tables define the relationships and the uncertainty of these variables. The variables are usually discrete with a fixed number of states. For each state, it gives the probability that the variable is in this state*". Figure 28 shows a hypothetical example of a BN with three variables: *The Number* *of contributor*s and *the Number of passive users* impact the *Size* quality subcharacteristic.

The structure of a BN is defined by a set of nodes and a set of edges for directed relationships between the nodes. Each node represents a variable with a set

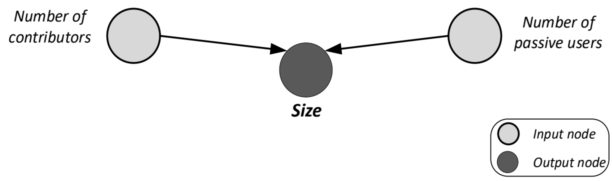

Figure 28. A basic example Bayesian network

of states or parameters, *"which are often discrete, mutually exclusive, and collectively exhaustive"* [157]. The arrows represent direct causal connections between the variables. *"Mathematically, the influences in the network are defined by conditional dependencies that are derived using probabilistic inference based on Bayes' theorem"* [158].

In order to illustrate the main features of BNs, a more detailed description of the BN is showed in Figure 29. For each node or variable there is a corresponding NPT. For example, the NPT of the node *Size* specifies that if the *Number of contributors* is *Low* and the Number of passive users is *High* the probability of *Poor* size is 80%. The CPD of the node Size has the values 47% and 53% for the parameters *Poor* and *Good* respectively. If the node has not parents, the NPT is not necessary to calculate the CPD.

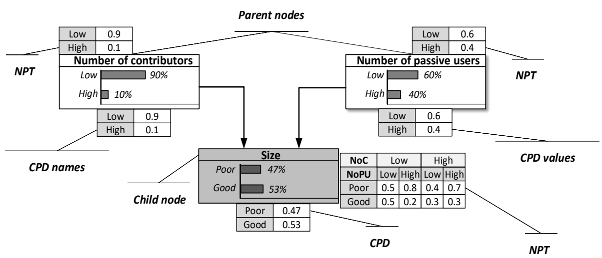

Figure 29. BN Hypothetical example

As shown in Figure 29, we specify for each node a *node probability table* (NPT). If a node has parent nodes, the NPT is calculated in dependence on the states of the parents. The problem of constructing NPTs is widely acknowledged in literature [159]. The solution of this problem involves defining quantitative relationships between variables. There are various methods for this quantification such as a probability wheel or regression from empirically collected data [119].

The CPD of each node is a conditional probability distribution obtained by applying the Markov property. This probability distribution, represented in a CPD, can be determined uniquely by the local NPT of each node [160].

4.2.4 Bayesian Inference

"Bayes' theorem provides a mechanism for updating the belief in some unknown hypothesis ***H*** *in light of new or additional evidence* ***E*** *(i.e., observed measures)"* [147]. For example, is the size of the OSSECO adequate? is the OSSECO effort well balanced? In other words, the theorem describes how to calculate the probability of ***H*** given the evidence ***E*** (i.e., ***P(H|E)***). What Bayes recognized was that it might not have direct information about ***P(H|E)*** but it does have prior information about ***P(E|H)***. The probability ***P(E|H)*** is called the likelihood of the evidence [147]:

$$P(H|E)=\frac{P(E|H)\times P(H)}{P(E|H)\times P(H)+P(E|\neg H)\times P(\neg H)}$$

4.2.4.1 Prior probabilities

The prior probability is the CPD over a variable X_i before any relevant evidence is obtained. It is related to historical datasets, expert judgment, or simply a function introduced for mathematical convenience [161]. For a variable X_i with not parents (i.e. an input variable) and with n parameters, its NPT is a single column with the discrete probability distribution over of the n parameters of X_i. On the other hand, for a variable X_k with parents (i.e. an output or intermediate variable) the NPT will have all possible combinations of parent states [162]. *There are several strategies for providing the prior probabilities of the input nodes:*

- Background information: these can be derived from the historical data of the variable. First, dividing the continuous range of historical data into a set of discrete and finite intervals (one interval for each state or parameter), this process is called discretization. Second, assigning probability values to each interval, according to the frequency distribution of the data in the interval (N. Fenton & Bieman, 2014).
- Experts: these can be provided by domain experts like in (Constantinou, Fenton, & Neil, 2016) and [164]. There are several strategies for eliciting expert knowledge like workshops, Delphi method and cross-impact analysis. The original Delphi experiment was conducted by Gordon and Helmer at the RAND Corporation around 1965 (Granger,1980). Figure 2 shows an approach for the method from Petrinja et al. [46]. The process begins with a set of properly worded questions given to each expert independently. Secondly, the answers are then summarized and shared amongst the group. Thirdly, the experts are asked to reconsider their responses in order to get a consensus with the responses of others. The sequence of this feedback and revision is continued until satisfied with their respective responses.

$$P(\theta_{ij}) = \frac{1}{Z_{ij}} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} \left(\sum \theta_{ijk} = 1, \theta_{ijk} \geq 0, \forall_k\right)$$

$P(\theta_{ij})$ is the prior probability of the parameter j of the variable i. The Z_{ij} is a normalization constant to ensure that $\int_0^1 P(\theta_{ij}) d\theta_{ij} = 1$. A hyperparameter α_{ijk} can be thought of as how many times the expert believes he or she will observe $X_i = k$ in a sample of α_{ij} examples drawn independently at random from distribution θ_{ij}.

4.2.4.2 Notation

In our approach, we define the BN to evaluate a QuESo-model subcharacteristic so that its corresponding measures are $M_1, M_2, \ldots, M_n$ as follows:

- Bayesian network (**BN**):

  $\boldsymbol{BN} = (G = (V, D), CPD)$

  $\boldsymbol{G} = DAG$ Direct Acyclic Graph

  CPD is a set of probability distribution indexed by a set of parameters ***U*** Over ***V***.
- Set of variables **(V)**:

$\boldsymbol{V} = (I \cup O)$ It is a set of nodes or variables.

$\boldsymbol{I} = (M_1, M_2, \ldots M_n)$ It is a set of ***n*** QuESo-model measures (input nodes).

$\boldsymbol{O} =$ It is the QuESo subcharacteristic (also called output node).

- Set of dependencies (**D**): It is a set of edges. It represents the direct dependencies between *I* and *O*.
- Set of parameters (**U**)
- $U_i=(S_1,S_2,\ldots,S_p)$ It is a set of ***p*** parameters or states with the probability distribution of the node X_i. Each S_i is a tuple <*label*, *value*>. Based on [147], the labels of the parameters for the ***I*** nodes are named: *Low*, and *High*, and the parameters for ***O*** nodes are named: *Poor* and *Good*.
- Set of node probability tables **(SNPT)**

  $SNPT(NPT_1, NPT_2, \ldots, NPT_w)$ For each node of the BN, there is a Node probability table (**NPT**).

4.2.4.3 BN learning

According *to Kosky & Noble, learning a BN is* "to estimate both the structure of the DAG (in other words, which directed edges should be used and which should be omitted) and, once the structure of the DAG has been established" [161], *the set of corresponding values of the CPD of each node. These tasks are done using historical data, classical learning algorithms like EM* [165], *BDeu+MLE, PC+MLE* [166] *or hybrid+MLE* [167], *and expert background knowledge about the specific domain.*

4.2.4.4 NPTs population

One of the hardest challenges in building a realistic BN model is to construct the NPTs. Because of this, it is common to incorporate expert knowledge to provide this type of information [163], [168]. However, in practice, the number of numerical parameters in a NPT grows exponentially. Consequently, "*obtaining large number of probabilities from domain experts is too expensive and too time-consuming in practice"* [169]. In order to solve this problem, a widely-accepted solution is the assumption of independence of causal influences (ICI). One of the most used ICI models is the *MaxNoisy* (similar to [147], [169]). This approach allows (for parametric models like BNs) using only a number of parameters that is linear instead of exponential to calculate the NPTs.

4.2.5 The QuESo Process

The idea to use BNs for evaluating software quality has been developed foremost by Fenton, Neil, and Littlewood in [147], [170] and [162]. They introduced BNs as *"a tool in software engineering and applied it in various contexts related to software quality"* [119]. Furthermore, there are several works that use BNs to evaluate the quality of software artefacts (e.g., [148], [154]). Our approach is closer to these works, primarily due to the hierarchical structure of the QuESo-model. BNs are a natural representation for the relationships between measures and quality subcharacteristics [154]. In this context, we divided the QuESo-process into two sub-processes. (1) Measurement process, to assess the available measures of the OSSECO. (2) Evaluation process, to establish a set of assessment goals from experts and to make the assessment of QuESo-model subcharacteristics.

4.2.5.1 Measurement process

The goal of this process is to obtain the input BN-nodes of the historical measures and the values of the of the absolute measure from the OSSECO. In order to do this, we need to collect the available data from OSSECO data sources, obtain the QuESo-model measures (i.e., historical and absolute measures), and configure the BNs to calculate the prior

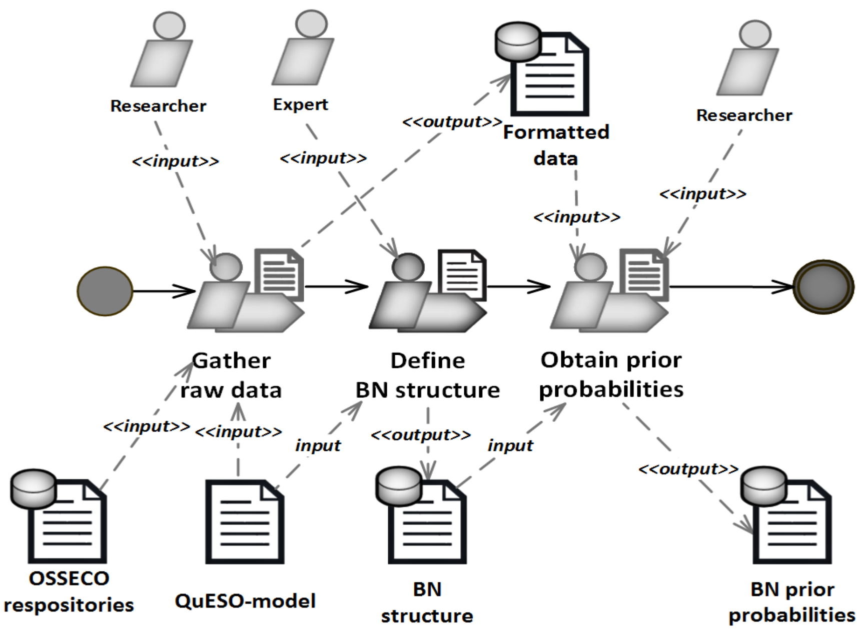

Figure 30. Measurement process

probabilities of the historical input nodes.

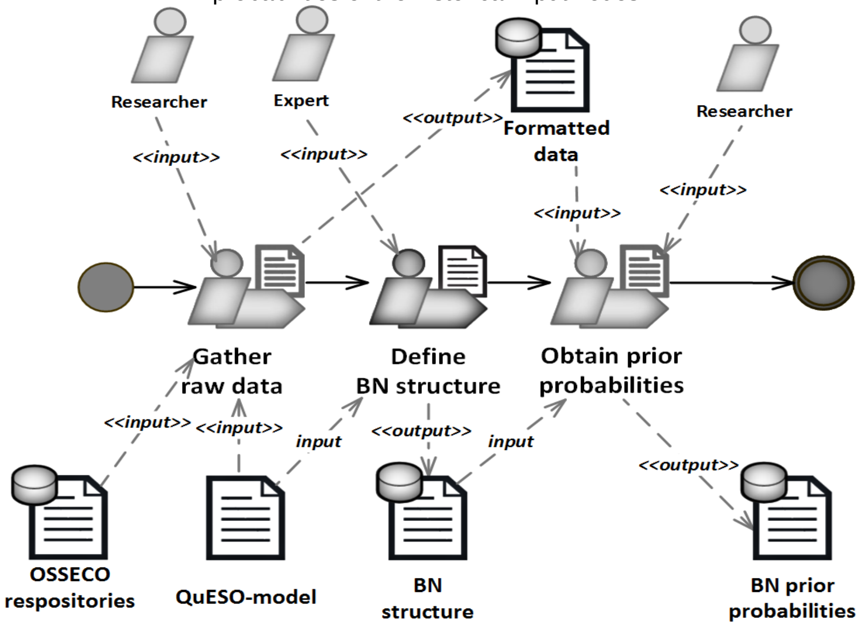

Figure 30 shows the phases of the process.

Gather raw data

The goal of this activity is to gather accurate and consistent measures of the OSSECO (see **Chapter 6** for details). The data is collected by:

- Implementing automatic instruments for gathering measure values from the OSSECO data sources (i.e., mailing lists, version control systems, bug tracker system, websites). Usually, OSSECOs provide open access to these data sources that can be accessed through web service interfaces (e.g., the JIRA bug tracking system provides a RESTful service).
- Applying automatic crawling to OSSECO websites.
- Surveying experts and OSSECO members.

Define BN structure

Building a BN involves two general tasks: the development of the network structure i.e., the variables and their relationships and the estimation of the parameters i.e., the CPD and NPT. For this study, we have defined the BN structure based on expert knowledge and literature. We designed a BN bi-graph with a d-connection topology. In other words, we did not apply any algorithm for learning the BN structure. This approach is widely applied in literature to calculate software measures [148], [147], [171].

- Khan et al., define a bi-graph as, *"if a node B is conditionally dependent on nodes A and C then, entering hard evidence at node A will update node B but will have no effect on node C"* [172]. It means that all measures have to be evaluated independently. Therefore, only the quality subcharacteristics have conditional dependences.
- In a converging d-connection topology, the network forms a directed causal graph, where edges go from causes (i.e., measures) to observable variables (i.e., QuESo quality subcharacteristics) [162].

Figure 31 depicts the topology of the BN defined for each of the QuESo-model subcharacteristics.

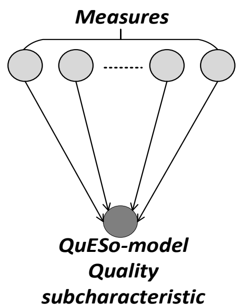

Figure 31. QuESo-model BN topology

Obtain BN prior-probabilities and absolute measures

In order to get the NPT prior-probabilities to the BNs, we classify the QuESo-model measures into two categories:

- ***Historical:*** Those that have evolution data over a time interval (e.g., *Number of Passive Users*, *Number of Commits*). The prior probabilities of this measures are assigned using the background information technique defined in Section 4.2.4.1.
- ***Absolute:*** Those that only have a single value (e.g., *Days After Last Commit*, *Cluster Coefficient*). The prior probabilities of these measures are provided by a group of domain experts (see Section 4.2.5.2).

Figure 32 shows an example of each type of measure.

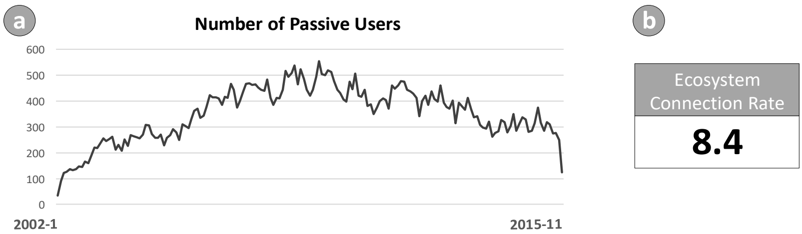

Figure 32. QuESo-model (a) historical measure (b) absolute measure

We implemented a set of computational tools in order to support the activities of this process (see Section 6).

4.2.5.2 Evaluation process

The goal to this process is to define a set of assessment scenarios in order to configure the QuESo-model for analyzing the data obtained from the measurement process. First, a set of experts configure the QuESo-model by providing the scenarios of the quality evaluation and selecting the QuESo-model subcharacteristics and measures that are relevant to reach those goals. Experts are also asked to provide the prior probabilities of the absolute measures. Furthermore, experts provide judgement about the validity of the input BN-nodes regarding to the assessment goals[19]. Figure 33 shows the phases of the process.

Configure model

The goal of this activity is twofold: first, to define a scenario for the evaluation and second, to assess the set of QuESo-model measures according to the defined scenario. This assessment should be based on the NPT prior-probabilities obtained in the assessment process.

[19] In https://sites.google.com/view/quesomodel, we designed a survey to configure the QuESo-model

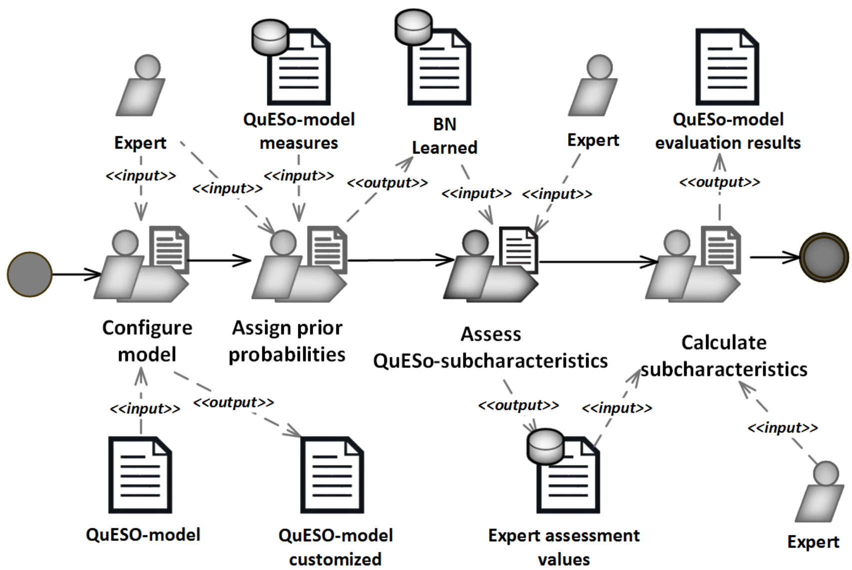

Figure 33. Evaluation process

Assign prior probabilities

In this activity experts provide the prior probabilities of the *Absolute Measures* that were calculated in the assessment process.

Judge BN input-nodes

The BN input-nodes calculated in the measurement process are judged by the experts in the context of the assessment goals. This evaluation will be the evidence to be used to compute the QuESo-model subcharacteristics.

Calculate subcharacteristics

In this activity, the outcome of each QuESo-model quality subcharacteristic is calculated based on the Bayesian inference algorithm [161]. Figure 34 shows the QuESo-tool web form with the *Size* subcharacteristic calculated. The figure shows the historical graph data of the measures, the Bayesian input nodes for each measure and the result of the assessment (i.e., BN output node calculated).

With the purpose of evaluating the usefulness of the QuESo process (i.e., "*The degree to which the subject considers that the evaluation process is effective in achieving its intended objectives*" [173]) in a real-world OSSECO, the QuESo process is validated by a case study on the Eclipse OSSECO (see Chapter 5).

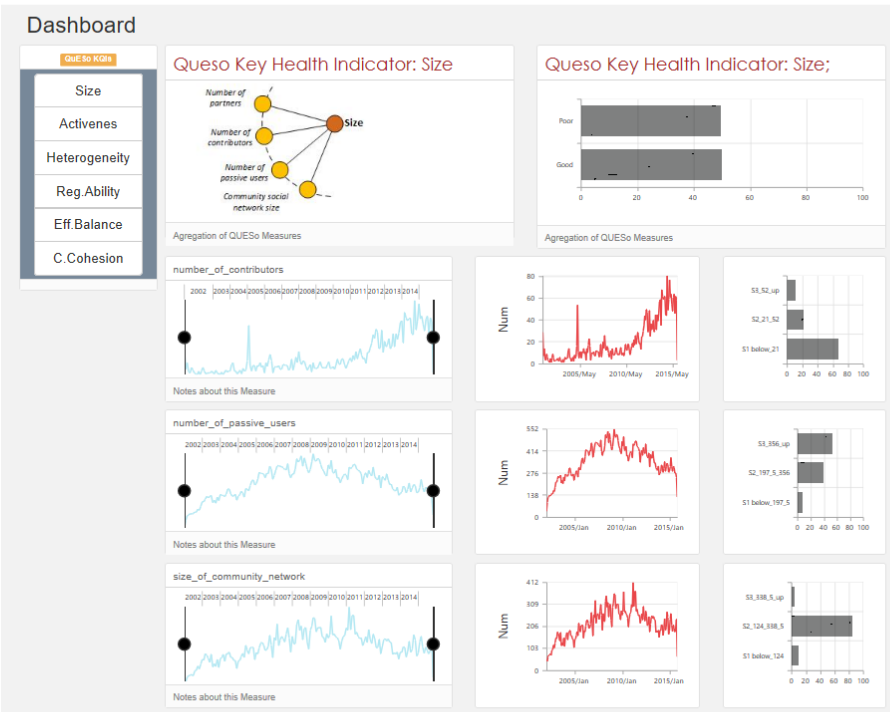

Figure 34. *Size* subcharacteristic calculated with the QuESo-tool

4.3 Conclusions

In this chapter, we have presented the QuESo-process, a process for conducting OSSECO quality evaluations using the QuESo-model. The QuESo-process has been divided in two sub-processes: (1) Measurement process to assess the available measures of the OSSECO. (2) Evaluation process to define a set of assessment goals and analyze the data from the measurement process.

The QuESo-process is based on building Bayesian networks (BN) as an evaluation model. These networks are derived systematically from: the QuESo-model, historical data from OSSECOs and empirical survey data from experts.

5 Framework Validation

"People tend to complicate their own lives, as if living weren't already complicated enough."
— **Carlos Ruiz Zafon**, **The Shadow of the Wind**

The validation of a framework that supports the quality evaluation of a system is a very important and a very difficult activity, in fact, it is not practically possible to specify or to measure all subcharacteristics for all parts of a software artifact [116], it also, requires long period of time. Similarly, it is not usually practical to specify or to measure quality in use for all possible stakeholder scenarios [174]. The framework should be tailored before using to identify those characteristics and subcharacteristics that are most important, and the different types of measure depending on the stakeholder goals and also to provide some evidence of the feasibility to obtain these measures, as mentioned in [113], one of the most habitual problems is the absence of data to calculate the measures.

In order to validate the QuESo-framework, we divided this chapter in two main sections. First, we provide a preliminary validation to evaluate the feasibility and availability of the QuESo-model measures using the GNOME OSSECO, GNOME community develops a free and popular desktop environment for GNU/Linux and UNIX-type operating systems. Second, we conducted a case study to evaluate the quality of Eclipse, an OSSECO with more than 300 projects and more than 200 partners. In the case study, we evaluate the validity of the QuESo-model results obtained after applying the QuESo-process and the usefulness and usability of the QuESo-tool for supporting the QuESo-framework.

5.1 GNOME Case

In this section, we present an early version of the QuESo-model validation. The goal of this validation is to provide evidence of the feasibility to obtain the measures, and consequently, the feasibility to evaluate the corresponding characteristics and sub characteristics proposed in QuESo. We hope to validate this feasibility using the QuESo measures identified in the literature related to the GNOME OSSECO.

We divided the process in two phases, similar to Samoladas [47]. First, the identification of the literature related to GNOME ecosystem. Second, the specification of the QuESo-model measures that are available for the GNOME ecosystem.

In the first phase, we have identified several papers that have measures for analyzing the GNOME ecosystem. In the second phase, the selected works were analyzed as follows: first, the measures with available values were extracted, secondly, the measures were classified according to the QuESo-model measure classification, and finally, we analyzed the situation for each quality aspects of QuESo-model.

Figure 35 shows the QuESo-model with the percentage of measures found in the literature for each quality aspect and Table 1 shows the number of measures found for each subcharacteristic. Table 47 shows a detailed table with the GNOME measures and the papers associated with it. The source column of this table lists the papers with values for each measure. The community quality dimension has measures associated with each subcharacteristic, and some of them have values for all the associated measures. However, in the case of the ecosystem network quality dimension, we have not found values for the measures related to resource health, and only the 40% of measures of network health have values.

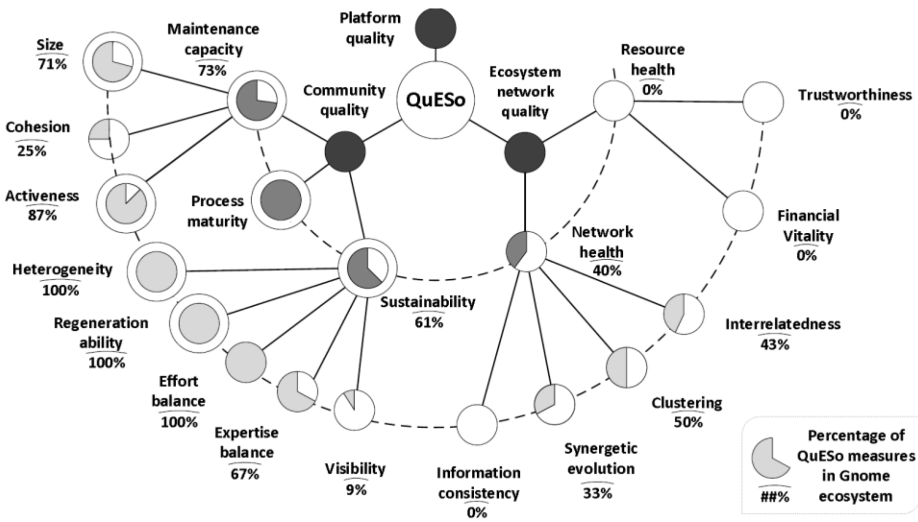

Figure 35. GNOME ecosystem evaluation with QuESo.

Table 46. Number of measures per quality subcharacteristics in GNOME literature

Subcharacteristic	Number of measures
Size	5
Cohesion	1
Activeness	13
Heterogeneity	1
Regeneration ability	2
Effort balance	11
Experience balance	2
Visibility	1
Information consistency	0
Synergetic evolution	1
Clustering	2
Interrelatedness	3
Trustworthiness	0

5.1.1 Conclusions

From the results shown in Figure 35 and Table 47. QuESo Measures in GNOME Ecosystem, we made some observations:

- There are many works with measure values for the community while there are few works with measure values related to the ecosystem network.
- The community quality dimension has measures associated with each subcharacteristic, and some of them have values for all the associated measures. However, in the case of the ecosystem network quality dimension we have not found values for the measures related to resource health, and only the 40%of measures of network health have values.
- The papers found do not cover the whole set measures in the QuESo quality model.

The first observation is also mentioned by other authors. For example, Jansen et al. [62] wrote that there is little literature that studied OSS from an ecosystem perspective, while Manikas et al. [142] wrote that most of the works studied OSS from a project or community perspective. For the second and third observation, we cannot state that there is full availability of measure values. However, it is worth mentioning that, in this case, we limited the sources to the ones published in the literature. Other methods can include direct access to the GNOME data sources (e.g., the number of files, number of downloads, web page request, and code vocabulary map). Other measures such as z-score, liquidity, and solvency can be obtained using expert surveys or mining public data

Table 47. QuESo Measures in GNOME Ecosystem

Measure	Sources
Bug tracking activity	R2, R3, R6, R11
Centrality	R6
Cluster of collaborating developers	R10
Community effort	R3
Contributor activity graph	R12
Contributor commit rate	R12
Date of last commit	R11
Developer activity diagrams	R10, R12
Distribution over the species	R11
Expertise view contributor	R12
Files changed	R12
Files per version	R6, R11, R12
Geographical distribution	R2
Lines added	R12
Lines changed	R12
Lines removed	R12
Mailing list	R2, R6, R11
Maximum Number of commits of a developer	R3, R12
Member effort	R10
Members activity rate	R11
New members	R6
Number community projects	R3
Number of active projects	R3, R10
Number of activity communities	R3, R7
Number of authors	R3, R11
Number of commiters	R3, R11
Number of commits	R3, R6, R10, R12
Number of contributors	R3, R12
Number of core developers	R6, R10
Number of developer projects	R12
Number of developer releases	R6
Number of mailing list users	R2, R6, R11
Number of members	R3, R6
Principal member activity	R3
Project activity diagrams	R2, R12
Project developer experience	R6
Temporal community effort	R3
Total effort of members	R3

5.2 A Case Study of Eclipse

In SECO research, a few proposals for quality evaluation of OSSECOs already exist, like [20], [45], [62], [120]. However, these proposals do not provide sufficient details to operationalize and lack evaluation process, for assessing software quality which limits their usefulness in practice.

This proposal attempts to fill the gap between OSSECO quality models and their operationalization. In order to do this, we use the QuESo model, described previously in Chapter 3, as a basis for quality evaluation of OSSECOs. It provides a structure of quality and comprehensive information about quality characteristics, subcharacteristics and measures.

In this chapter, we present a case study of QuESo-framework, to perform an evaluation of quality subcharacteristics of the Eclipse-OSSECO. In order to do this, we applied the QuESo-process explained in **Chapter 4.** Furthermore, this process is supported by the QuESo-tool (see **Chapter 6**) in order to calculate quality subcharacteristics based on Bayesian Networks (BN) approach.

5.2.1 Case Study Protocol

In this case study, we combined an exploratory approach [175], [176] with Technical Action Research (TAR) from [1]. Both methods are useful for gaining a deeper understanding of a specific phenomenon (i.e., OSSECO quality) in its natural context (i.e., a real-world OSSECO).

This section describes the protocol of the case study according to the guidelines proposed by [177]. The activities of the protocol are summarized in Figure 36

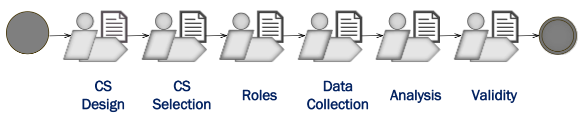

Figure 36. Case study protocol

5.2.1.1 Case Study Design

The purposes of this activity are: (i) to clarify the reasons for undertaking this study, (ii) to identify the context of the case study (i.e., single case or multiple case studies) and to describe the object of study, (iii) to define the goals, research questions and metrics to be addressed in this study.

- **Rationale**

This case study was undertaken in response to the lack of operationalization of OSSECOs quality models, in order to be used in the evaluation and prediction of quality. Several authors pointed out the important challenge of evaluating OSSECO quality from an ecosystem point of view [39], [62], [119]. Furthermore, this case study attempts to address research challenges published by authors such as Jansen, (2014), who promoted in the research community the need to provide more insight for the measures and methods related to OSSECOs and Manikas, (2016) who published a call to action for more in-depth than in-width SECO studies.

• Context and object of study

We decided to undertake a single exploratory case study [176], which involved a real OSSECO (Eclipse) and an OSSECO quality model (QuESo- model). The object of study is the QuESo-framework.

• Goals, research questions and metrics

We used Goal Question Metric approach (GQM) from Basili et al., (2009) and Basili & Weiss, (1984), to identify the overall goal to be achieved by the case study. Furthermore, we derived a set of questions for this goal. Finally, we analyzed each question to identify the metrics we needed. The combined answers to these questions helped to determine whether our goals were being satisfied.

According to Wieringa, (2014) we classified the research as a Knowledge Goal (**KG**)

- **KG:** To validate the feasibility of using the QuESo-framework for evaluating the quality subcharacteristics of a real-world OSSECO.

We used the three research questions (defined in Section 1.4) in order to structure the case study design.

- **RQ2.1:** What is the perceived validity of the QuESo-model? (i.e., "The degree to which all the elements contained in a quality model should actually appear in the model in the right way" [173]).
- **RQ2.2**: What is the validity of the results obtained applying the QuESo-process?
- **RQ2.3:** What is the perceived usefulness of the QuESo-process? (i.e., "The degree to which the subject considers that the evaluation process is effective in achieving its intended objectives" [173]).

The last step in the GQM approach is to define a set of measures in order to answer the research questions adequately. Table 48 shows all derived metrics related to the RQs.

Table 48. Metrics associated to the RQs

Metric	Description
RQ2.1: What is the validity of the QuESo-quality model?	
M1.1	**Perceived QuESo-model validity (Over the QuESo quality model subcharacteristics).**
	The answers ranged on a five-point Likert scale from *"Not important" to "Very important"* (subjective, ordinal data)
M1.2	**Percentage total of QuESo measures that can be calculated automatically**
	Ratio between # automatically-collected measures and number total of measures of the QuESo-model (ratio scale)
RQ2.2: What is the validity of the results obtained applying the QuESo-process?	
M2.1	**Perceived validity of the results obtained applying the QuESo-process**
	The answers ranged on a five-point Likert scale from *"Strongly disagree" to "Strongly agree"* (subjective, ordinal data)
RQ2.3: What is the perceived usefulness of the QuESo-process?	
M3.1	**Perceived usefulness of the QuESo-process**
	The answers ranged on a five-point Likert scale from *"Strongly disagree" to "Strongly agree"* (subjective, ordinal data)

5.2.1.2 Case Study Selection

Taking into account the goal and the research questions of this case study, several OSSECOs were potential candidates to be selected. In this section, we discuss the case selection strategy and describe the general characteristics of the OSSECO selected.

Selection strategy

We considered the following criteria to select the case study: (i) the public availability of the information related to the OSSECO (e.g., code repositories, mailing lists, bug trackers); (ii) domain experts and members of the community willing to contribute to this study; (iii) the size of the OSSECO. Using these criteria, we selected Eclipse as the OSSECO because of this, we had the opportunity to extract useful data from the various data sources that constitute the OSSECO.

5.2.1.3 The Eclipse Case

Eclipse is one of the most successful OSSECO, in recent years, from several perspectives (i.e., project, community and ecosystem perspectives). Eclipse is an exemplary case of OSSECO [180]. Eclipse is formed by a network of: software projects, software communities, users, providers, common platform, open data sources and relationships with other OSSECOs [139]. According to [181], Eclipse is divided in two main connected components:

- Eclipse platform: *"to provide vendor-neutral, open development platform supplying frameworks and exemplary, extensible tools"*.
- Eclipse network: "*to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services".*

The keystone player of the Eclipse OSSECO is the Eclipse foundation, which purpose is *"to advance the creation, evolution, promotion, and support of the Eclipse Platform"* [182]. Furthermore, it organizes a number of activities (e.g., marketing events, community conferences), and it does not set the directions of the Eclipse OSSECO.

5.2.1.4 Case Study Roles

Based on the TAR roles described in Wieringa, (2014), we defined three main roles:

- *Stakeholders* (Wieringa helper role): In our approach, we identified three different stakeholders in which an actor wanted to know the quality of an OSSECO: (1) potential adopters: organizations or individuals who are interested in using the projects produced by an OSSECO; (2) potential contributors: individuals who are interested in collaborating in some form with the OSSECO (e.g., coding, documenting, reporting bugs); Finally, (3) OSSECO members: authors, adopters, passive users, etc., who are interested in the OSSECO quality assurance.
- *Experts* (Wieringa empirical researcher role): The experts answer some validation knowledge question about the use of the QuESo-framework for evaluating Eclipse quality. We identify two types of experts: domain experts (i.e., experts knowing the Eclipse context) and methodology experts (i.e., experts knowing how to use the QuESo-process).
- *Researchers* (Wieringa technical researcher role): The group of researchers that designed the QuESo-framework.

5.2.1.5 Empirical approach

In order to collect the data related to the metrics defined in Table 48, we designed a set of three survey questionnaires. We decided to use descriptive survey-questionnaires because, as stated by Linåker et al., [55], they give the researcher support to make claims or assertions about the population (e.g., What are the experts or stakeholders' view on the new OSSECO quality model?). Furthermore, we used web-based questionnaires because "*they are more time efficient and help to acquire higher response rate than mailed questionnaires. They are easy to set up and then their distribution is also very simple and straightforward by sending the corresponding link to target audience. Data collection by investigators is also easy because it does not require the time-consuming data entries*"[55] .

Survey design

As described by Wohlin et al., [111], the objective of a survey is to provide a "snapshot" of the current status related to a phenomenon. In order to do this, surveys collect qualitative and quantitative information that usually is obtained employing questionnaires. To ensure rigor and repeatability and to reduce researcher bias, the survey protocol was designed following the survey methodological approach described by Linåker et al., [55]. We designed three questionnaires for covering the three research questions (see Table 48). (1) First questionnaire was designed to measure the perceived QuESo-model validity (i.e., the RQ2.1). The **RQ2.1** survey consisted of 16 questions (one for each QuESo-model subcharacteristic). Each question was rated on a 5-point Likert scale where 1= *Unimportant* and 5=*Very Important*. (2) Second questionnaire was designed to measure the perceived validity of the results obtained applying the QuESo-process (i.e., the RQ2.2). Similar to the RQ2.2 survey, The **RQ2.2** survey consisted of 16 questions (related with QuESo-model subcharacteristics). Each question was rated on a 5-point Likert scale where 1=*Strongly Disagree* and 5=*Strongly Agree*. Finally, (3) Third questionnaire was designed to measure the perceived usefulness of the QuESo-process (i.e., the RQ2.3). The **RQ2.3** survey consisted of 5 questions. Each question was rated on a 5-point Likert scale where 1=*Strongly Disagree* and 5=*Strongly Agree*. The questionnaires may be publicly accessed[20].

[20]https://goo.gl/forms/SlOfVsufU4EXtjWM2

https://goo.gl/forms/UeWwbXmhpwYoDRrk1

https://goo.gl/forms/HQJxXhT30T7GHJX53

Survey evaluation

All survey questionnaires were evaluated using expert reviews. Experts helped correct survey questionnaire flow, check the wording used in the questionnaire and align the survey instrumentation with the case study's main objective. As a result of this evaluation, some changes were implemented in the survey questionnaires.

Channel

The survey was implemented using Google Forms®, which offers support to develop Internet questionnaires and collecting and managing their data.

Population

The theoretical population was divided into three sub-populations related with each questionnaire. For the first questionnaire, the population was IT professionals and researchers with experience in OSS communities. A specific set of experts in OSSECOs, Eclipse measures, OSS communities and Bayesian networks for evaluating software quality indicators were the population for the second questionnaire. Finally, for the third questionnaire, the population was IT professionals related with the Eclipse OSSECO.

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 [183]. 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 [184]. These groups were the following: members of the RISCOSS project with experience in SECOs, experts in OSSECOs, OSS communities and Bayesian network analysis, Eclipse OSSECO members.

5.2.1.6 QuESo-validity study (RQ2.1)

First, we present the results of the survey on perceived QuESo-model validity **(M1.1)**. Next, we provide the study results about percentage total of QuESo measures that can be calculated automatically **(M1.2).**

Participants

We had a total of 15 participants from different countries; 7 from Spain, 6 from Colombia and 2 from Mexico. 12 of the participants had Ph.D. degree and 3 had master degrees. Regarding their organizations, 11 of the respondents were working in universities (e.g., Politécnica de Catalunya, Nacional de Colombia, Industrial de Santander) and 4 were working in the industry (e.g., Hewlet Packard, Eng-House).

Perceived QuESo-model validity Survey:

The alternative hypotheses are shown in Table 49. Alternative hypotheses correspond to our expectation: the importance of QuESo-model subcharacteristics for evaluating the quality of OSSECOs. Null hypotheses refer to the very little importance or no importance of the QuESo-model subcharacteristics for evaluating the quality of OSSECOs.

In order to analyze the results of the RQ.2.1 survey, we used the procedures defined by Boone and Boone for Likert-type data (i.e., statistical median for *Central tendency*, frequencies for *Variability* and *Wilcoxon signed rank* for hypotheses testing) [185].

Table 49 Specification of RQ2.1 hypotheses

Alternative Hypothesis id	Description
$H_{A2.1.1}$	Agree that *Size* is important (i.e., $H_{A1.1}$: $\mu(M1.1) > 3$; $H_{01.1}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.2}$	Agree that *Activeness* is important (i.e., $H_{A1.2}$: $\mu(M1.1) > 3$; $H_{01.2}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.3}$	Agree that *Expertise* Balance is important (i.e., $H_{A1.3}$: $\mu(M1.1) > 3$; $H_{01.3}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.4}$	Agree that *Heterogeneity* is important (i.e., $H_{A1.4}$: $\mu(M1.1) > 3$; $H_{01.4}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.5}$	Agree that *Regeneration Ability* is important (i.e., $H_{A1.5}$: $\mu(M1.1) > 3$; $H_{01.5}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.6}$	Agree that *Effort Balance* is important (i.e., $H_{A1.6}$: $\mu(M1.1) > 3$; $H_{01.6}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.7}$	Agree that *Visibility* is important (i.e., $H_{A1.7}$: $\mu(M1.1) > 3$; $H_{01.7}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.8}$	Agree that *Community Cohesion* is important (i.e., $H_{A1.8}$: $\mu(M1.1) > 3$; $H_{01.8}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.9}$	Agree that *Ecosystem Cohesion* is important (i.e., $H_{A1.9}$: $\mu(M1.1) > 3$; $H_{01.9}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.10}$	Agree that *Information Consistency* is important (i.e., $H_{A1.10}$: $\mu(M1.1) > 3$; $H_{01.11}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.11}$	Agree that *Synergetic Evolution* is important (i.e., $H_{A1.11}$: $\mu(M1.1) > 3$; $H_{01.11}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.12}$	Agree that *Interrelatedness Ability* is important (i.e., $H_{A1.12}$: $\mu(M1.1) > 3$; $H_{01.12}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.13}$	Agree that *Niche Creation* is important (i.e., $H_{A1.13}$: $\mu(M1.1) > 3$; $H_{01.13}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.14}$	Agree that *OSSECO Knowledge* is important (i.e., $H_{A1.14}$: $\mu(M1.1) > 3$; $H_{01.14}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.15}$	Agree that *Vitality* is important (i.e., $H_{A1.1}$: $\mu(M1.1) > 3$; $H_{01.15}$: $\mu(M1.1) \leq 3$).
$H_{A2.1.16}$	Agree that *Trustworthiness* is important (i.e., $H_{A1.16}$: $\mu(M1.1) > 3$; $H_{01.16}$: $\mu(M1.1) \leq 3$).

Central Tendency: As shown in Figure 3, the perceived importance of the QuESo-model subcharacteristics was, in general, high (median = 4.0).

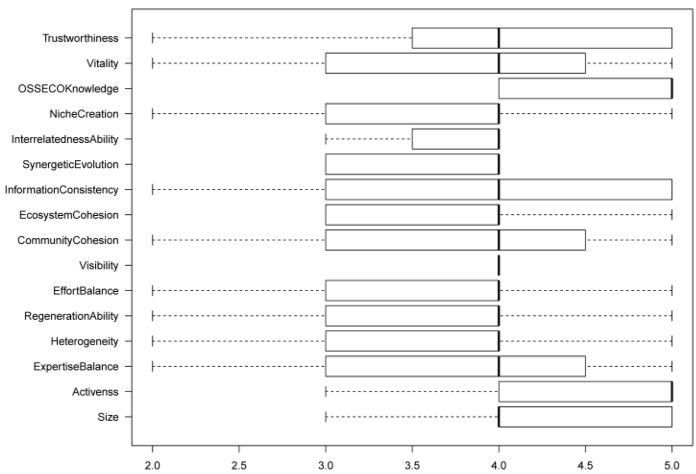

Figure 37 Importance of QuESo-model quality subcharacteristics. (1= *Unimportant* and 5=*Very Important*)

Furthermore, all of the evaluations of OSSECO Knowledge and Activeness were higher than 3. This analysis contributes to answering RQ1

Variability: The descriptive statistics, see Figure 38, provide the distribution of answers according to the type of Likert scale items. This support the tendency of the perceived importance of the QuESo-model subcharacteristics. Only Synergetic evolution was evaluated as not important for one of the subjects. This analysis contributes to answering RQ1.

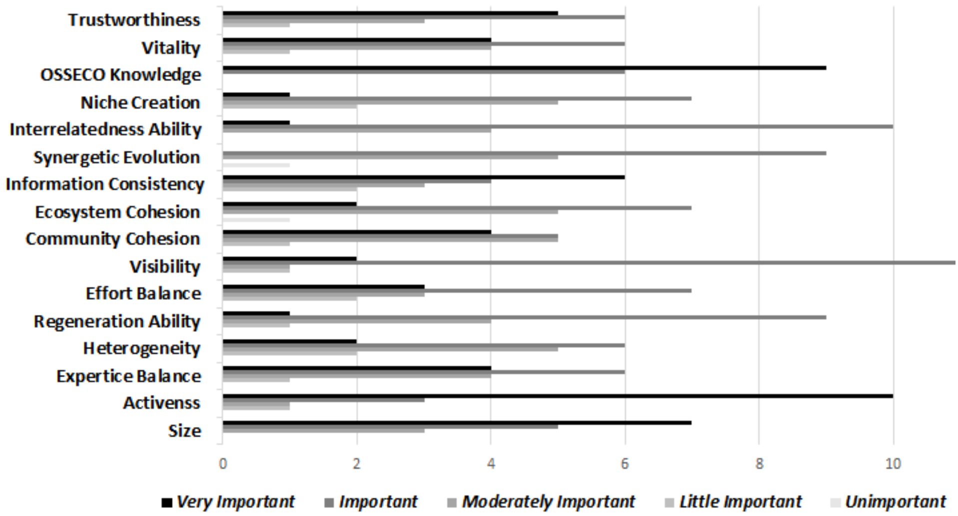

Figure 38 Frequency of answers per QuESo-model subcharacteristic

Hypotheses test: Similar to Kläs et al., all hypotheses were tested with a one-tailed Wilcoxon signed rank test with a significance level of ***p=0.05***. Table 50 shows ***p-value<=0.5*** for all hypotheses. Thus, the null hypotheses $H_{01.*}$ are rejected and the alternative hypotheses $H_{A1.*}$ are corroborated, demonstrating that the subject perceived the importance of the QuESo-model subcharacteristics for evaluating the quality of OSSECOs. This analysis contributes to answering RQ2.1.

Table 50 P-values of hypotheses

Alternative Hypothesis id	Description
$H_{A2.1.1}$	*Size, accepted, p-value = 0.000296*
$H_{A2.1.2}$	*Activeness, accepted, p-value*= 0.000232
$H_{A2.1.3}$	*Expertise* Balance, accepted, p-value=0.000316
$H_{A2.1.4}$	*Heterogeneity,* accepted, p-value= 0.0003144
$H_{A2.1.5}$	*Regeneration, Ability* accepted, p-value= 0.0002587
$H_{A2.1.6}$	*Effort Balance,* accepted, p-value= 0.0003079
$H_{A2.1.7}$	*Visibility* accepted, p-value= 0.0001994
$H_{A2.1.8}$	*Community Cohesion,* accepted, p-value= 0.0003201
$H_{A2.1.9}$	*Ecosystem Cohesion,* accepted, p-value= 0.0002984
$H_{A2.1.10}$	*Information Consistency,* accepted, p-value= 0.0003201
$H_{A2.1.11}$	*Synergetic Evolution,* accepted, p-value= 0.0002516
$H_{A2.1.12}$	*Interrelatedness Ability,* accepted, p-value= 0.000228
$H_{A2.1.13}$	*Niche Creation* accepted, p-value= 0.0002984
$H_{A2.1.14}$	*OSSECO Knowledge* accepted, p-value= 0.0002413
$H_{A2.1.15}$	*Vitality* is important accepted, p-value= 0.000316
$H_{A2.1.16}$	*Trustworthiness,* accepted, p-value= 0.0003128

Percentage total of QuESo measures that can be calculated automatically

As Figure 39 shows, most of the QuESo-model measures of the Eclipse OSSECO can be calculated automatically (58; 65%). 20 of the measures are available in different Eclipse resources (e.g., Eclipse web site). Only 12 of the 90 measures of the Eclipse-OSSECO were not available. Table 57 shows the list of available measures calculated by the QuESo-tool. This analysis contributes to answering RQ2.1.

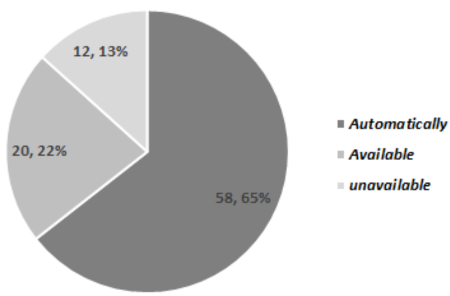

Figure 39 Distribution of QuESo-measures for availability

5.2.1.7 QuESo-validity of the results study (RQ2.2)

Similar to the validation of the perceived QuESo-model validity survey, we specified analyzing the results of this survey using statistical median and Wilcoxon *signed rank test* for testing a set of alternative hypotheses (See Table 55).

Participants

In order to answer RQ2.2 we identified the top developers of the Eclipse OSSECO[21]. It was interesting to ask participants their opinion about the validity of the results obtained applying the QuESo-process because developers are one of the most important keystone actors in a OSSECO [186]. Although the survey was sent to the top-100 Eclipse developers, only five developers completed the survey. However, the developers in our sample have different professional profiles and they are from several countries. Table 51 shows the characteristics of the Eclipse developers.

Table 51 Characteristics of Eclipse developers

Expert	OSSECO Background
A	Software Engineer from Germany.
B	Software Developer from Canada
C	Software Architect from Germany.
D	Software developer at Ericsson Hungary
E	Software Developer form Italy

Perceived validity of the results obtained applying the QuESo-process

In order to measure the perceived validity of the results obtained applying the QuESo-process, participants were asked for their perception of each of one of the values calculated for the 16 QuESo-model quality subcharacteristics in a resilient context. The alternative hypotheses are showed in Table 52. Alternative hypotheses correspond to our expectation: the validity of the results obtained applying the QuESo-process to the Eclipse OSSECO.

[21] The list of the top-100 developers of the Eclipse OSSECO is available in http://dashboard.eclipse.org

Table 52 Specification of RQ2.2 hypotheses

Alternative Hypothesis id	Description
$H_{A2.2.1}$	Agree that *Size* result is valid (i.e., $H_{A2.1}$: $\mu(M1.1) > 3$; $H_{02.1}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.2}$	Agree that *Activeness* result is valid (i.e., $H_{A2.2}$: $\mu(M1.1) > 3$; $H_{02.2}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.3}$	Agree that *Expertise* Balance result is valid (i.e., $H_{A1.3}$: $\mu(M1.1) > 3$; $H_{02.3}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.4}$	Agree that *Heterogeneity* result is important (i.e., $H_{A2.4}$: $\mu(M1.1) > 3$; $H_{02.4}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.5}$	Agree that *Regeneration Ability* result is valid (i.e., $H_{A2.5}$: $\mu(M1.1) > 3$; $H_{02.5}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.6}$	Agree that *Effort Balance* result is valid (i.e., $H_{A2.6}$: $\mu(M1.1) > 3$; $H_{02.6}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.7}$	Agree that *Visibility* result is valid (i.e., $H_{A2.7}$: $\mu(M1.1) > 3$; $H_{02.7}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.8}$	Agree that *Community Cohesion* result is valid (i.e., $H_{A2.8}$: $\mu(M1.1) > 3$; $H_{02.8}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.9}$	Agree that *Ecosystem Cohesion* result is valid (i.e., $H_{A2.9}$: $\mu(M1.1) > 3$; $H_{02.9}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.10}$	Agree that *Information Consistency* result is valid (i.e., $H_{A2.10}$: $\mu(M1.1) > 3$; $H_{02.10}$: $\mu(M1.1) \leq$ 3).
$H_{A2.2.11}$	Agree that *Synergetic Evolution* result is valid (i.e., $H_{A2.11}$: $\mu(M1.1) > 3$; $H_{02.11}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.12}$	Agree that *Interrelatedness Ability* result is valid (i.e., $H_{A2.12}$: $\mu(M1.1) > 3$; $H_{02.12}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.13}$	Agree that *Niche Creation* result is valid (i.e., $H_{A2.13}$: $\mu(M1.1) > 3$; $H_{02.13}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.14}$	Agree that *OSSECO Knowledge* result is valid (i.e., $H_{A2.14}$: $\mu(M1.1) > 3$; $H_{02.14}$: $\mu(M1.1) \leq 3$).
$H_{A2.2.15}$	Agree that *Vitality* result is valid (i.e., $H_{A2.1}$: $\mu(M1.1) > 3$; $H_{02.15}$: $\mu(M1.1) \leq 3$).

Central Tendency: As shown in Figure 40, the perceived validity of the results obtained applying the QuESo-process to evaluate the quality of the Eclipse OSSECO was, in general, high.

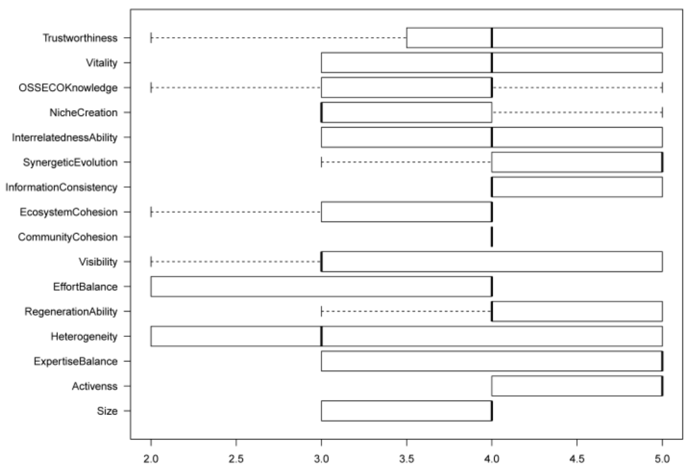

Figure 40 Validity of QuESo-process results. (1= Strongly disagree and 5=Strongly agree)

Hypotheses test: All hypotheses were tested with a one-tailed Wilcoxon Table 53signed rank test with a significance level of ***p=0.5***. shows ***p-value*** calculated for all hypotheses. This analysis contributes to answering RQ2.2.

Table 53 P-values of hypotheses

Alternative Hypothesis id	Description
$H_{A2.1.1}$	*Size, accepted, p-value = 0.02667*
$H_{A2.1.2}$	*Activeness, accepted, p-value*= 0.02667
$H_{A2.1.3}$	*Expertise* Balance, accepted, p-value=0.02667
$H_{A2.1.4}$	*Heterogeneity,* accepted, p-value= 0.02838
$H_{A2.1.5}$	*Regeneration, Ability* accepted, p-value= 0.02838
$H_{A2.1.6}$	*Effort Balance,* accepted, p-value= 0.02667
$H_{A2.1.7}$	*Visibility* accepted, p-value= 0.02838
$H_{A2.1.8}$	*Community Cohesion,* accepted, p-value= 0.02724
$H_{A2.1.9}$	*Ecosystem Cohesion,* accepted, p-value= 0.02724
$H_{A2.1.10}$	*Information Consistency,* accepted, p-value= 0.02838
$H_{A2.1.11}$	*Synergetic Evolution,* accepted, p-value= 0.02724
$H_{A2.1.12}$	*Interrelatedness Ability,* accepted, p-value= 0.02838
$H_{A2.1.13}$	*Niche Creation* accepted, p-value= 0.02724
$H_{A2.1.14}$	*OSSECO Knowledge* accepted, p-value= 0.02895
$H_{A2.1.15}$	*Vitality* is important accepted, p-value= 0.02838
$H_{A2.1.16}$	*Trustworthiness,* accepted, p-value= 0.002724

5.2.1.8 QuESo-usefulness study (RQ2.3)

Similar to the validation of the perceived QuESo-model validity survey, we specified analyzing the results of this survey using statistical median and Wilcoxon *signed rank test* for testing a set of alternative hypotheses (See Table 55).

Participants

The QuESo-framework provides a natural mechanism for combining knowledge from experts and stakeholders with quantitative data. We decided to survey the five experts that configured the QuESo-model (see Section 4.2.5) to determine the usefulness of the QuESo-process. Table 54 shows the characteristics of the experts.

Table 54 Characteristics of experts

Expert	OSSECO Background
A	Co-founder of the company that provides the dashboard and analytics services to the Eclipse-OSSECO
B	Senior researcher in statistics and applied mathematics. Co-founder of a company specialized in data analytics, with several publications in applied Bayesian Networks.
C	Senior researcher in Software Ecosystems and Open Source Software
D	Senior researcher in Software Ecosystems and Open Source Software
E	Keystone developer in an OSS-community

Perceived usefulness of the QuESo-process

Hypotheses described in Table 55 were based on the *technology acceptance model* (TAM[22]) questionnaire [187], a well-evaluated measurement instrument for these type of characteristics [118].

Table 55 Specification of RQ2.3 hypotheses

Alternative Hypothesis id	Description
$H_{A2.3.1}$	QuESo-model could accomplish the evaluation of OSSECOs more quickly (i.e., $H_{A2.3.1}$: $\mu(M1.1) >3$; $H_{02.3.1}$: $\mu(M1.1) \leq 3$).
$H_{A2.3.2}$	QuESo-model could make OSSECO-evaluation job of easier (i.e., $H_{A2.3.2}$: $\mu(M1.1) >3$; $H2.3_{02.3.2}$: $\mu(M1.1) \leq 3$).
$H_{A2.3.3}$	QuESo-model could improve the quality of the results of the OSSECO evaluation (i.e., $H_{A2.3.3}$: $\mu(M1.1) >3$; $H_{02.3.3}$: $\mu(M1.1) \leq 3$).
$H_{A2.3.4}$	QuESo-model could improve the effectivness of the evaluation (i.e., $H_{A2.3.4}$: $\mu(M1.1) >3$; $H_{02.3.4}$: $\mu(M1.1) \leq 3$).
$H_{A2.3.5}$	Agree that *Regeneration Ability* is important (i.e., $H_{A2.3.5}$: $\mu(M1.1) >3$; $H_{02.3.5}$: $\mu(M1.1) \leq 3$).

[22] The TAM questionnaire states six questions with respect to the perceived usefulness

Central Tendency: As shown in Figure 41, the perceived usefulness of the QuESo-process was, in general, high (median = 4.0).

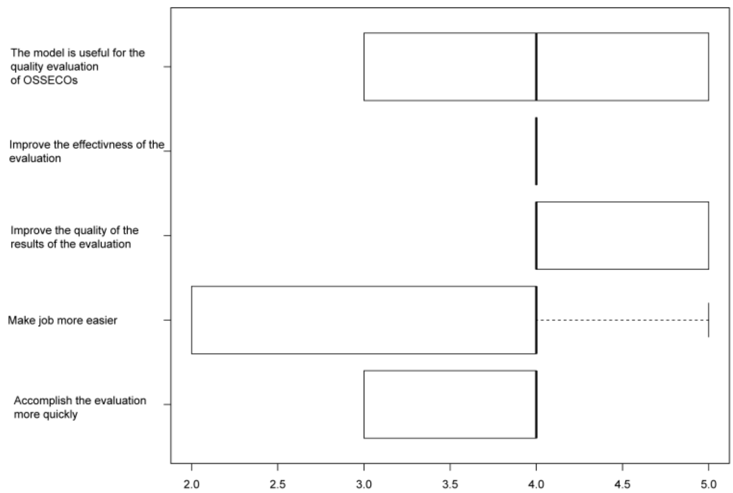

Figure 41 Importance of QuESo-model quality subcharacteristics. (1= Strongly disagree 5=Strongly agree)

Hypotheses test: All hypotheses were tested with a one-tailed Wilcoxon signed rank test with a significance level of ***p=0.05***. Table 50 shows ***p-value*** calculated for all hypotheses. This analysis contributes to answering RQ2.3.

Table 56 P-values of hypotheses

Alternative Hypothesis id	Description
$H_{A2.3.1}$	*Accepted, p-value = 0.02667*
$H_{A2.3.2}$	*Not accepted, p-value= 0.02838*
$H_{A2.3.3}$	*Accepted*, p-value= 0.02838
$H_{A2.3.4}$	*Not Accepted*, p-value= 0.02724
$H_{A2.3.5}$	*Accepted*, p-value= 0.02838

5.2.1.9 Conclusions

In this section, we have presented a preliminary analysis of the results of a case study in which the main goal was to investigate the validity of the QuESo-framework for evaluating the quality of the Eclipse-OSSECOs. The case study is a survey-based study that analyzes the perception of participants about the validity and usefulness of the QuESo-framework. We designed three surveys, one for each RQ2 subquestions. The main findings of these surveys reveal that the QuESo-framework is perceived as a good mechanism for evaluating the quality of OSSECOs. The case study also shows that more of the QuESo-model measures are available to be used by the QuESo-framework. The measures that were not available, were related to financial information (e.g., ZETA score of partners, market share, number of markets, etc.)

6 QuESo Tool

"Theory is the practice of the impotent."
— **Carlos Ruiz Safon**, **The Angel's Game**

The quality evaluation of OSSECOs is influenced by many factors, including the size of the ecosystem, the health of the network, the financial vitality of its partners, the sustainability of the OSSECO community and all those defined in the QuESo-model. In addition, the operationalization of the OSSECO quality models is a complex and often underestimated problem that has previously been poorly treated. This chapter presents the QuESo-tool, a set of tools to support semi-automatic quality evaluation of OSSECOs. The chapter is divided in two sections: first, it describes an early attempt to validate the QuESo-tool. It is a software prototype named SALMonOSS for monitoring OSSECO measures. Second, it describes a set of software components that support the QuESo-process in order to evaluate the quality of OSSECOs.

6.1 SALMonOSS

The motivation of this contribution is to provide a Proof of Concept (PoC) of the QuESo-tool that allows gathering, monitoring and visualization of data from OSSECOs repositories. In addition, this PoC is useful to check the viability of the QuESo-tool platform architecture (**RQ2**). As a PoC, we designed a tool which adopts the techniques from monitoring Quality of Service (QoS) to monitor the OSS community health.

In this section, we present a software prototype which adopts the techniques from monitoring Quality of Service (QoS) of services to monitor the OSSECO community health. By following analogue principles, we argue that a set of techniques and applications based on top of the monitored data can be also ported from SOC to OSSECO.

The implementation of the prototype was focused on monitoring mailing lists, bug tracking systems and version repositories. Particularly, the technologies chosen are Markmail, Jira and GIT, respectively. All these technologies were required to be used as services.

6.1.1 Related Work

6.1.1.1 Assessment of OSS community health

The assessment of OSS community health is usually realized by tools for a particular community or for a specific platform. For instance, there exist several solutions in the literature for monitoring and analysis of specific OSS communities by accessing directly to their available data repositories, e.g. Nagios [128], GNOME [28], Ruby [188], Debian [86], Eclipse [186] and Sourceforge [189]. Generally, these tools are specifically designed for one or a few scientific experiments [28].

Consequently, these tools are not available once the research is finished and are not suitable to monitor other communities.

On the other hand, there are frameworks and tools that automate the monitoring and analysis over specific repositories, e.g. FLOSSMETRICS (flossmetrics.org), QUALIPSO (qualipso.org), QUALOSS (qualoss.eu), the ongoing OSSMETER (ossmeter.eu), LTC (passion-lab.org/projects), MARKOS (markosproject.eu), Ohloh (ohloh.net) and finally Goeminne and Mens [28]. Each work introduces a generic and extensible framework for studying OSS communities. Although these tools are more generic and reusable, most of the presented works are limited to specific technologies (e.g. Bugzilla, CVS, Git, Jira) that are monitored separately. Hence, the analysis is performed without a clear picture of the system as a whole (e.g. an OSS community may decrease the number of forum posts because they tend to use the mailing list more and not because the community shrinks).

6.1.1.2 Quality assessment in SOC

In the field of SOC, there are several works addressing the challenges of monitoring the QoS of services and the analysis of the compliance of their Service Level Agreements (SLAs), which is a document that states the levels of acceptance with respect to the QoS. The automation of monitoring has been already achieved by different works (either by gathering the data using online testing [190], passive monitoring [191] or a combination of both). Most of the monitoring solutions are domain-independent, and can be used for different services regardless of the domain. The analysis of the SLAs has also been addressed in depth, eventually providing a comprehensive explanation of the cause of SLA violations [192].

To the best of our knowledge, there is only one work, from Ghezzi et al. [193], that applies the idea of implementing data repositories of an OSS into services. In this work, they present a framework named SOFAs, that implements the different repositories as RESTful services to monitor the evolution of OSS. However, their work is restricted to software evolution, and the analysis performed is not suitable to check the OSS community health.

6.1.2 Aligning OSS and SOC

6.1.2.1 OSS Health

Key Performance Indicators (KPIs) can determine the health of an ecosystem. For example, Iansiti and Levien have introduced three determinants KPIs of business ecosystem health: robustness, productivity and niche creation [115]. The concept Key X Indicator is not only related to performance. Ferguson et al. used the KPI for quantifying uncertainty in early lifecycle cost estimation [164], Raanan and Kenett used Key Risk Indicators (KRIs) for monitoring risks in business unit [194]. Furthermore, KRIs were used in the RISCOSS Analytics tool for the development of a risk management approach applicable to the adoption and deployment of OSS [195]. In this chapter, we have defined the concept Key Health Indicators (KHIs) as a valid instrument to evaluate the OSS community health. This is inspired by a Software Engineering Institute (SEI) approach called QUELCE [195] for handling early stage project planning estimation. Consequently, KHIs may support strategic decision making within the OSS community and the OSS adopter. To this aim, KHIs should be continuously monitored to evaluate the OSS community health.

The list of health indicators may be very large. Therefore, the KHIs should be reduced to an efficient set of indicators that capture the impact on the OSS community health. To do this, in first place, values of OSS community variables are extracted from data and metadata saved in OSS community repositories (e.g., version control systems, mailing lists, bug trackers, websites, wikis, discussion forums). Secondly, social network data is generated using social network analysis tools (e.g., NodeXl [196]). Subsequently, KHIs are calculated using Bayesian networks by fusion of community variables and domain expert knowledge to support the definition of relations between OSS community measures and KHIs [195] (e.g., number of commits and activeness of the community).

We have defined an earlier architecture for extracting and processing OSS data, based on the 3-layered approaches proposed by Franch et al. [197] and Geommine and Mens [28] (See Figure 43). Layer-I is for extracting raw data from OSS community repositories. In layer II, the measurements are processed and structured, e.g. in terms of the member relationships collected from community social media. This social data is gathered via natural language processing techniques. In layer III, the KHIs are calculated using Bayesian networks and OSS-expert assessment based on their experience in OSS communities.

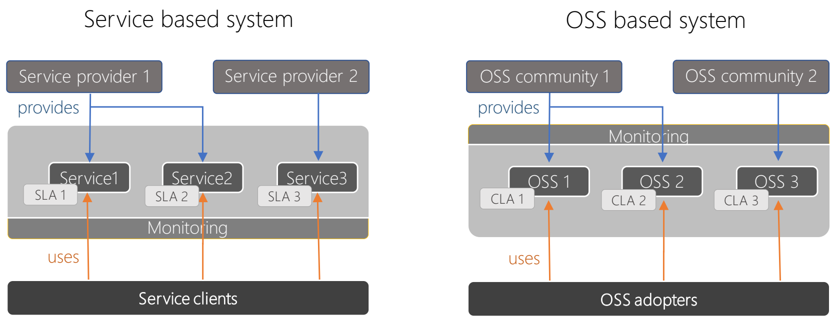

Figure 42 Figure OSS and service based systems aligned.

6.1.2.2 Alignment with SOC

Figure 42 shows graphically how we substantiate the idea of aligning SOC and OSS. In the field of SOC, illustrated in the left-hand side of the figure, services are provided by service providers, who must guarantee a certain quality of the service (QoS) as to fulfill the client's needs about performance, availability, etc., which are operationalized into an SLA. An SLA is composed of a set of Service Level Objectives (SLOs), which constrain the permissible values that service metrics (throughput, response time, etc.) may take. The QoS is computed in a regular basis using a monitor, which measures the behavior of the services during their execution. The monitoring results are checked against the SLA by an analyzer, and possible violations are detected and reported. On top of this basic schema, several tasks as service selection [198], and techniques as proactive adaptation [199], may benefit from the monitoring infrastructure. In the right-hand side of the figure, we may see that OSS communities are the counterpart of service providers, since they distribute the OSS component outside. OSS adopters (the counterpart of service clients) need to operationalize the quality that they demand to the OSS component in terms of what we may call Component Level Objectives (CLO) which together conform a Component Level Agreement (CLA).

As said above, in this validation we focus on those CLOs that involve metrics that are bound to the information gathered from the community through resources like bug trackers and forums (mean time to repair a bug, volume of messages per day, etc.). These metrics can be aggregated to evaluate and assess KHIs (like timeliness and activeness). On top of this scenario, several tasks as OSS selection, and techniques as social network analysis, may benefit from this monitoring schema.

6.1.2.3 SALMonOSS

To prove the feasibility of the approach, we have implemented a framework named SALMonOSS. SALMonOSS is an OSS Health monitoring framework based on an existing technology named SALMon [53] developed in our research group.

SALMon is a versatile monitoring framework to monitor the QoS of services. Here we describe how we have aligned the problem of assessing OSS community health with service monitoring, and the enhancements of SALMon to support OSS monitoring, resulting in the SALMonOSS framework.

6.1.2.4 General requirements

The process of monitoring the OSS community health has a set of high-level requirements that must be addressed to properly develop a monitoring solution for OSS. Based on the different aspects of the monitoring process in SOC and their applicability to the OSS domain, we have identified the set of requirements that are briefly described below.

Servitization

OSS development and maintenance usually requires different software management tools, such as mailing lists, bug tracking systems and version repositories; each one providing an aspect of the OSS community health. *Requirement 1*. OSS management tool functionalities shall be offered as services. Options are: 1) the tool already provides a web service interface (e.g., the JIRA bug tracking system provides a RESTful service), 2) the management tool is wrapped into a web service (see Figure 44). By monitoring services instead of components, OSS community health can be assessed by current service monitoring technologies with minimal changes.

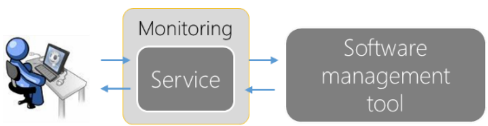

Figure 44 Wrapping a tool into a service.

Strategies

SOC provides two strategies to conduct monitoring, active monitoring and passive monitoring. Requirement 2. In order to not lose the potential of SOC monitoring, SOC strategies need to be applicable in OSS community health analysis. The meaning needs to be slightly adapted, though:

- Active: Under this approach, monitoring can be performed without the involvement of the OSS community. A software component invokes periodically a set of operations of the services that retrieve the status of the OSS. By monitoring the results of these invocations, the metrics related to the OSS community health are computed. This is analogous to active monitoring (or online testing) in SOC.
- Passive: Under this approach, OSS communities are required to manage the development and maintenance of the OSS using the wrapping services (e.g.

any new bug is reported using the service interface). Transparently to the service client, services are continuously monitored and whenever a new event occurs, the metrics related to the OSS community health are calculated. This is analogous to passive monitoring in SOC.

It is worth to remark that these approaches are not mutually exclusive and they can be combined. That is, when monitoring the health of an OSS community, some data can be retrieved using the passive approach and the rest of the data using the active approach.

Extensibility

It is difficult to predict in advance the full set of metrics that may need to be monitored. We can expect that this may be even more difficult in the future since the field of OSS community health analysis still has room for improvement. Requirement 3. OSS community health analysis tools need to be extensible to host new metrics.

Interoperability

In SOC, monitoring tools are the basis for applying more sophisticated treatments: self-adaptation, prediction, etc. This potential should not be lost when adapting to the OSS field. Requirement 4. OSS community health analysis tools need to support interoperability with other, potentially unknown, tools. This way, we can think of supporting tasks like assessing in the decision of going or not for an update in a given moment, or to decide to start risk mitigation actions when some community indicator goes beyond some threshold.

CLA-aware

We have already explained the importance that SLA has in SOC monitoring. SLAs are a way to express the expectations of end users on the monitored system, and this concept seems also useful for our OSS context. Requirement 5. OSS community health analysis tools need to be customized in a particular deployment according to the contents of the CLA. Therefore, the tool will just monitor the metrics that are required to check the CLA, which is the ultimate goal of the monitoring process.

6.1.2.5 Monitoring process

SALMonOSS supports both passive and active approaches to monitor the required services (Req 2). To combine both approaches (see Figure 45), the Invoker (1a) uses the same infrastructure as used by the OSS community (1b). In both cases, the invocations are performed through the same Enterprise Service Bus (ESB), which is a middleware able to handle and manage the communication between services. When receiving a service request, the ESB notifies to the Monitor the event (2). The Monitor then notifies it to the Measure Instruments that are responsible to handle such request (3). In parallel to the previous monitoring activities, the ESB forwards the service request to the Software management tool, who may have been wrapped if it is not providing a service itself (Req 1) (4). The service invokes the software management tool (5), and receives the response (6). The response is sent through the ESB (7) to the initiator of the request, either the Invoker or the OSS community (8). In parallel, the same response is notified to the Monitor (9), which notifies it to the Measure Instruments (10), which ultimately calculate the measurements (11). By following this approach, new metrics can be calculated by implementing the required measure instruments.

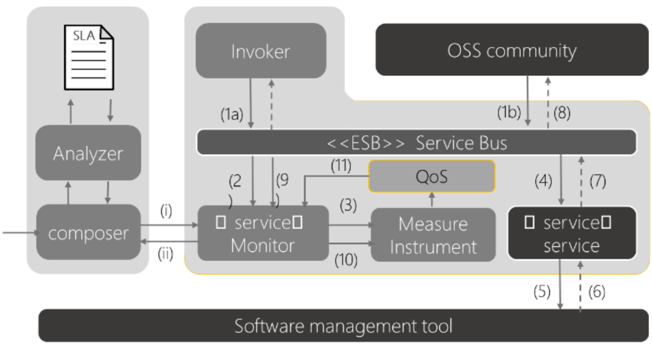

Figure 45 OSS monitoring process in SALMonOSS.

SALMon has been implemented as a service based system by itself, and so does SALMonOSS. By following the SOA principles, SALMonOSS can be easily integrated to different frameworks that require monitoring (Req 4), as happened with SALMon that was used to support self-adaptive service based systems [200], federated cloud management [201], or service selection [198], among others. On the other hand, being the monitor service an aggregate of measure instruments, it satisfies Req 3: measure instruments are pluggable components that deal with a particular metric each, so that the monitor instantiates and manages the required set of measure instruments.

Finally, SALMonOSS can be automatically configured to monitor the metrics included in a CLA, with the combination of the ADA software (Req 5). ADA is an Agreement Document Analysis framework aimed at extracting useful information from agreement documents [192], and with SALMon constitute a combined framework named SALMonADA. Under this framework, the CLA is parsed by the Composer, and the Monitor is configured through its interface (i). Whenever new measurements are computed, they are reported to ADA through the composer (ii) in order to perform the analysis of the CLA fulfillment.

For example, with SALMonOSS it is possible to monitor metrics of the OSS community such as commits per day or time to resolve a bug. When a new commit or bug is captured by the ESB (either by passive or active approach), the Measure Instruments compute the metrics, which are then analyzed against the CLA to check if there is any violation over the conditions which would compromise the KHIs. In such a case, the interested parties are notified.

6.1.2.6 Implementation and preliminary results

The current implementation of the prototype is focused on monitoring mailing lists, bug tracking systems and version repositories. Particularly, the technologies chosen are Markmail, Jira and GIT, respectively. All these technologies are required to be used as services. Jira already provides a RESTful service interface, whereas Markmail and GIT tools have to be wrapped. With respect to the GIT tool, we have implemented a service that wraps the GIT with the methods to retrieve the details related to commits. Preliminary results of monitoring the GIT repository using the active monitoring strategy are shown in Figure 46. We have monitored the behaviour of the commits performed, which is an indicator for the activeness of the community on the xwiki1 during the month of December. Particularly we have monitored the following metrics: number of commits per day, files changed per day, lines added per day, lines removed per day, average files changed per commit, average lines added per commit and average lines removed per commit. As shown, we can observe that the community is active during weekdays, but inactive during weekends and holidays.

All the retrieved information is then checked against the CLA, in order to identify possible deviations and eventually assess their KHIs.

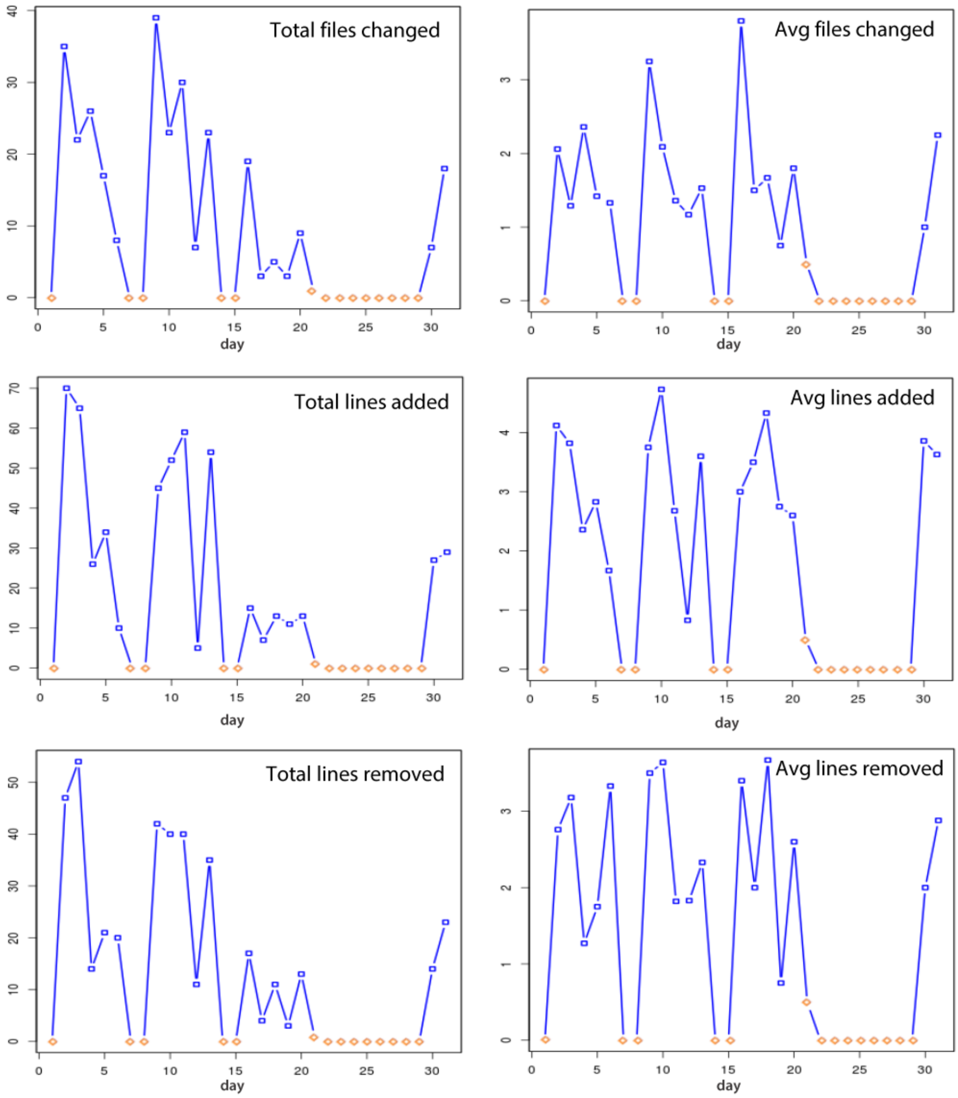

Figure 46 Metrics related to GIT monitored with SALMonOSS.

6.2 QuESo-tool system

The QuESo-tool system is the technological support of the QuESo-process, helping OSSECO researchers and experts to operationalize the QuESo-model in order to evaluate the quality of OSSECOs. By using this software system, we expected to:

- Support OSSECO stakeholders and actors for the quality evaluation process of OSSECOs.
- Allow OSSECO monitoring by implementing a set of REST services that could be consumed for the other software application.
- Provide researchers with a source of measures and indicators for analyzing OSSECOs.

The use case diagram of the system is shown in Figure 47.

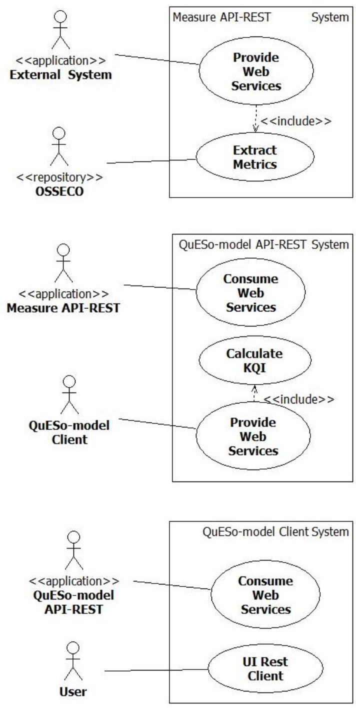

Figure 47 Use case diagram of the QuESo-tool system

The researchers, experts and external stakeholders in general may access to the quality evaluation of the OSSECOs through the QuESo-model client system.

Furthermore, they could access the individual Key Quality Indicators (KQI)[23] through their own software applications if these applications implement access to the QuESo-model API REST services.

The goal of the Measure API REST is to give an example of the RESTful service, that OSSECOs have to provide, in order to use the QuESo-framework for evaluating their quality.

6.2.1 Overall Architecture

The QuESo-software tool, schematically shown in Figure 48, was built according to a layered architecture style and based on SOA principles.

- OSSECO server: this layer represents the OSSECO, it implements several tools for gathering the data from the OSSECO data sources (i.e., code version repositories, bug trackers, mailing lists, web pages and OSSECO community members, data bases and other dashboard APIs). The implementation is based on SOA (Service Oriented Architecture). This system provides a JSON/REST API, implemented in Java, which expose the values of the Historical and Absolute measures of the OSSECO.

  The *Measure Extractor* component is a set of Java classes that gather the data from Eclipse repositories and crawler the Eclipse OSSECO web sites.

- QuESo-framework client/server: this is the core component of the QuESo-tool. It implements all of the functionalities necessary to support the reasoning process of the BNs (i.e., define the BN structure, calculate BN prior probabilities and calculate QuESo-model subcharacteristics). In order to do that, we developed a Java application using the jSMILE library [202]. In addition, the application provides a JSON/REST API which expose the values of the Bayesian network CPD calculated.
- QuESo-framework client: In order to support the validation of the results obtained by using the QuESo-framework. This component implements a web dashboard that summarizes the QuESo-model subcharacteristics evaluated[24]. This component implements the design pattern Model-View-Controller (MVC). This system is developed in C#.

---

[23] At the beginning of the project, we used the term *Key Health Indicator* (KHI) because, this was the term used by several authors for assessing OSSECO characteristics (e.g., [62], [142]). However, we decided to use *Key Quality Indicators* (KQI) rather than KJHI, because KQI is a broader term. (e.g., QuESo-model include *Health* as one of his quality subcharacteristics).

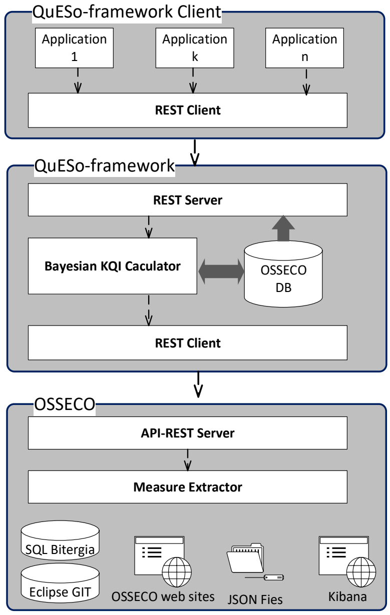

Figure 48 QuESo-tool architecture

6.2.1.1 QuESo-tool for gathering OSSECO measures

This section shows how to use the JSON/REST API to query the Eclipse OSSECO measures. All API-REST is accessible from the URI http://testoneosseco.azurewebsites.net/.

The response is sent as JSON. Figure 49 shows the data received for the GET command querying the absolute measure *Number of Partners*:

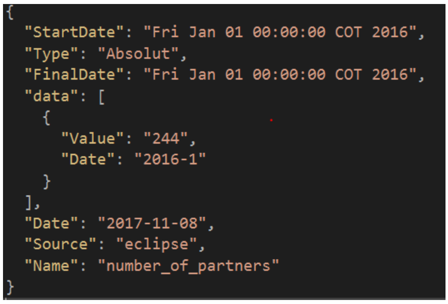

Figure 49 OSSECO REST-API Response

Table 57 shows the list of the OSSECO RESTful services.

6.2.1.2 QuESo-tool for gathering QuESo-model subcharacteristics

The use of the JSON/REST API to query the QuESo-model subcharacteristics (i.e., *KQIs*) is similar to the described above. Figure 50 shows the response to the GET for *Size KQI*.

Figure 50 QuESo-framework REST-API Response

The list of all QuESo-framework RESTful services is showed in Table 57.

Table 57 List of OSSECO RESTful services

/OSSECOMeasuresRESTServer/measures/numberofpartners
/OSSECOMeasuresRESTServer/measures/numberofcontributors
/OSSECOMeasuresRESTServer/measures/numberofpassiveusers
/OSSECOMeasuresRESTServer/measures/sizeofcommunitynetwork
/OSSECOMeasuresRESTServer/measures/versionhistory
/OSSECOMeasuresRESTServer/measures/numberofopenedbugs
/OSSECOMeasuresRESTServer/measures/numberofclosedbugs
/OSSECOMeasuresRESTServer/measures/daysafterlastcommit
/OSSECOMeasuresRESTServer/measures/daysafterlastrelease
/OSSECOMeasuresRESTServer/measures/numberofchangedfiles
/OSSECOMeasuresRESTServer/measures/numberoffilesincommits
/OSSECOMeasuresRESTServer/measures/numberofmessagessent
/OSSECOMeasuresRESTServer/measures/numberofsentresponses
/OSSECOMeasuresRESTServer/measures/numberofcommits
/OSSECOMeasuresRESTServer/measures/numberofcommiters
/OSSECOMeasuresRESTServer/measures/communitycommitrate
/OSSECOMeasuresRESTServer/measures/numberofevents
/OSSECOMeasuresRESTServer/measures/culminatingpoint
/OSSECOMeasuresRESTServer/measures/decliningpoint
/OSSECOMeasuresRESTServer/measures/timeliness
/OSSECOMeasuresRESTServer/measures/timezone
/OSSECOMeasuresRESTServer/measures/memberactivitytypes
/OSSECOMeasuresRESTServer/measures/numberOfProjectTypes
/OSSECOMeasuresRESTServer/measures/communitybetweennesscentralityrate
/OSSECOMeasuresRESTServer/measures/clustercoheficient
/OSSECOMeasuresRESTServer/measures/degreeofkeystoneactors
/OSSECOMeasuresRESTServer/measures/clustersofcollaborators
/OSSECOMeasuresRESTServer/measures/nodestodisconnectOSSECO
/OSSECOMeasuresRESTServer/measures/ecosystemdegreeofkeystoneactors
/OSSECOMeasuresRESTServer/measures/ecosystemclustercoheficient
/OSSECOMeasuresRESTServer/measures/ecosystemconnectionrate
/OSSECOMeasuresRESTServer/measures/clusteringbytermsimilarity
/OSSECOMeasuresRESTServer/measures/ecosystemage
/OSSECOMeasuresRESTServer/measures/upperquartileofcommits
/OSSECOMeasuresRESTServer/measures/lowerquartileofcommits
/OSSECOMeasuresRESTServer/measures/medianofcommits
/OSSECOMeasuresRESTServer/measures/meanofcommits
/OSSECOMeasuresRESTServer/measures/maximumnumberofcommits
/OSSECOMeasuresRESTServer/measures/minimumnumberofcommits
/OSSECOMeasuresRESTServer/measures/newmembers
/OSSECOMeasuresRESTServer/measures/newcontributors
/OSSECOMeasuresRESTServer/measures/contributorssurvivalrate
/OSSECOMeasuresRESTServer/measures/membersmakingnewfeaturerequests
/OSSECOMeasuresRESTServer/measures/numberofnaturallanguagessupported
/OSSECOMeasuresRESTServer/measures/numberofmailinglistsubs
/OSSECOMeasuresRESTServer/measures/totalnumberofdownloads
/OSSECOMeasuresRESTServer/measures/longevityofcontributorrate
/OSSECOMeasuresRESTServer/measures/numberofbacklinks
/OSSECOMeasuresRESTServer/measures/numberofwebhits
/OSSECOMeasuresRESTServer/measures/ossecoprofit
/OSSECOMeasuresRESTServer/measures/varietyofOSSECOpartners
/OSSECOMeasuresRESTServer/measures/proyectcommittimerate
/OSSECOMeasuresRESTServer/measures/communityactivityrate
/OSSECOMeasuresRESTServer/measures/subcommunitiesmembersrate
/OSSECOMeasuresRESTServer/measures/numberofscientificpublications
/OSSECOMeasuresRESTServer/measures/numberofpatents
/OSSECOMeasuresRESTServer/measures/authorratingandreputation
/OSSECOMeasuresRESTServer/measures/contributorsexpertiserate
/OSSECOMeasuresRESTServer/measures/projectspercontributorrate
/OSSECOMeasuresRESTServer/measures/numberofOSSECOknowledgeartifacts
/OSSECOMeasuresRESTServer/measures/varietyinOSSECOtechnologies
/OSSECOMeasuresRESTServer/measures/numberofnichesprojects
/OSSECOMeasuresRESTServer/measures/partnercontributionrate
/OSSECOMeasuresRESTServer/measures/OSSECOpopularity
/OSSECOMeasuresRESTServer/measures/OSSECOpartnersclosenesscentrality
/OSSECOMeasuresRESTServer/measures/OSSECOpartnersbetweennesscentrality
/OSSECOMeasuresRESTServer/measures/OSSECOpartnersconnectedness
/OSSECOMeasuresRESTServer/measures/OSSECOreciprocity
/OSSECOMeasuresRESTServer/measures/activityperiod
/OSSECOMeasuresRESTServer/measures/numberofactivitytypes

Table 58 List of QuESo-framework RESTful services

eNIe	
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Size
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Activeness
	/PLATEOSSRESTServer/bayesian/bayesianqueso/ExpertiseBalance
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Heterogeneity
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Regeneration Ability
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Effort Balance
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Visibility
	/PLATEOSSRESTServer/bayesian/bayesianqueso/CommunityCohesion
	/PLATEOSSRESTServer/bayesian/bayesianqueso/EcosystemCohesion
	/PLATEOSSRESTServer/bayesian/bayesianqueso/InformationConsistency
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Synergetic Evolution
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Interrelatedness Ability
	/PLATEOSSRESTServer/bayesian/bayesianqueso/NicheCreation
	/PLATEOSSRESTServer/bayesian/bayesianqueso/OSSECO Knowledge
	/PLATEOSSRESTServer/bayesian/bayesianqueso/Vitality

The Bayesian networks generated by both components can be exported as XDSL files. This kind of files can be opened by using GeNIe[25], a tool for modeling and learning with Bayesian networks. Figure 51 shows a BN for the *Visibility* QuESo-model subcharacteristic.

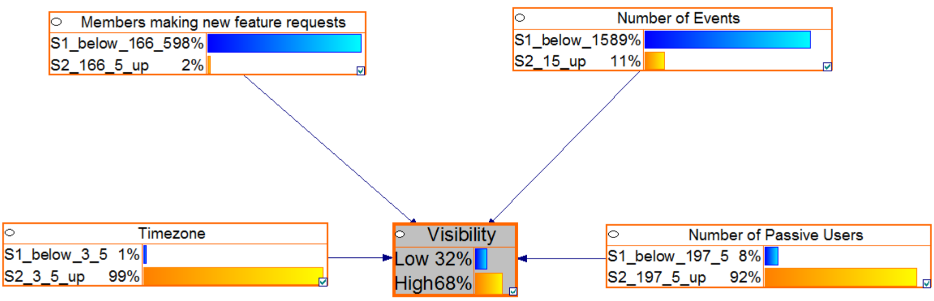

Figure 51 BN for the Visibility QuESo-model subcharacteristics

[25] https://www.bayesfusion.com/

6.2.1.3 QuESo-tool for visualizing the quality subcharacteristics

In order to visualize the set of measures and quality subcharacteristics gathered from the OSSECOs we develop a web application to visualize the QuESo-model subcharacteristics and measures (i.e., historical and absolute). Figure 52 shows an example of the *Heterogeneity* KQI. The figure shows the Bayesian nodes calculated for each historical measure, the historical variation of the data and the Heterogeneity KQI calculated value. This application can be accessed at http://plateoss.azurewebsites.net/

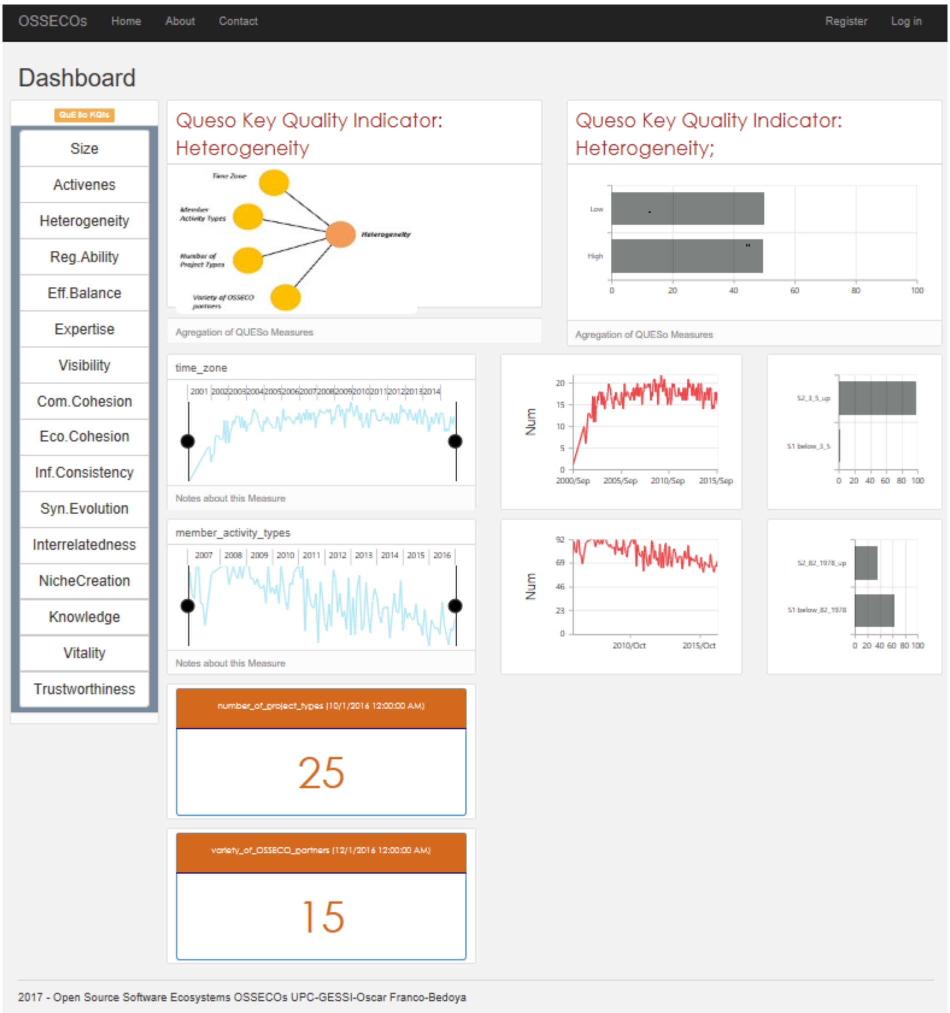

Figure 52 QuESo-framework web application

6.3 Conclusions

This chapter is divided twofold. First, we demonstrate how current SOC monitoring techniques can be ported and used to assess the OSS community health. Particularly, we have presented SALMonOSS, an OSS community health monitoring framework able to: (1) monitor a list of community health metrics along time (2) link the gathered values with client's needs by operationalizing conditions in CLAs and (3) engineer a portfolio of methods and techniques that supports OSS communities and OSS adopters (e.g. OSS selection, proactive adaptation, etc.). Furthermore, the solution follows a layered architecture that structures the data from low-level metrics to KHIs. Second, we present the set of software components that support the QuESo-framework for evaluating the quality of OSSECOs. QuESo-tool is a SOA based applications that: (1) expose a JSON/RESTful interface for data accessing of the OSSECO measures. (2) expose another JSON/RESTful interface for getting the Bayesian nodes calculated for the OSSECO measures and the calculated KQIs and (3) a web application that consumes this REST/JSON services.

6.3.1 Applications

Applying the same paradigm from SOC to OSS and the SOA based architecture puts forward the capability to transfer and reuse the knowledge and techniques from one field to the other. Particularly interesting are those techniques which have a straightforward resemblance on their objectives. We describe below some clear applications:

OSS selection

As a first step for OSS adoption, a particular OSS has to be selected. There are different OSSs that fulfill the same functional requirements for an OSS adopter, hence the selection of an OSS should be based not only on the functional requirements but also on the quality and sustainability of the OSS, which is directly related to the health of the OSS community. The selection process can be facilitated in an automated manner through technologies that implement algorithms based on multiple-criteria decision analysis. There are several frameworks that implement those algorithms in the field of service selection [198], which we argue could be applied in the field of OSS.

OSS violation prediction

During the execution of the OSS, violations of the CLA can be forecasted before they occur by applying analysis over statistical models using the monitored data. The results of such forecasting can be reported to the OSS community, who in turn, can apply the required mechanisms to mitigate any risk and avoid such violations (e.g. reallocating resources and assigning new priorities). There exist different techniques in the field of SOC that predict violations of SLAs [200], we believe that the same techniques could be applied to predict violations of CLAs for OSS.

OSS adaptation

In case of a CLA violation, the OSS adopter might be interested on performing an adaptation to reestablish the indicators specified in the CLA. For instance, if there's a critical security bug on an OSS component without a clear commitment by the OSS community to solve it in a short period of time, the OSS adopter might replace or disable the OSS until the issue is solved. In the field of SOC, automatic adaptation of services has been a major topic of study. In SOC, when a SLA is violated, an adaptation need is triggered, which prompts the generation of an adaptation plan that is ultimately enacted by a component able to execute the adaptation [203]. Although service and OSS adaptation have their differences in terms of technology and standardization, the progress and results made could be reused and applied in the field of OSS.

7 Conclusions

"It seems that, in the advanced stages of stupidity, a lack of ideas is compensated by an excess of ideologies."
— **Carlos Ruiz Safon**, **The Angel's Game**

In this chapter we present the conclusions of this thesis by answering the research question that were stated in Chapter 1.

> **RQ1:** What is currently known about the characteristics of OSSECOs?

In order to answer this question, we conducted a systematic mapping study in the fields of open source software (OSS) and software ecosystems (SECOs) (see Chapter 2). The interest on OSS and SECOs stems from the fact that they are two consolidated research areas in software engineering. We designed and followed a rigorous protocol which uncovered up to 87 papers from a gross total of 652, to answer the different research sub questions that we defined. We summarize these answers below:

SM-RQ1. What are the demographic characteristics of the studies about OSSECOs?

We divided this research question in several sub questions in order to identify: (1) the type of papers, in particular empirical, is important because they are indicators of the maturity in a new research field; (2) the evolution in the number of publications. It is an indication of how the activity of a research field changes; (3) the information about geographical distribution of the publications; (4) the classification between academy and industry. It is relevant because OSSECOs concepts extend geographical and institutional boundaries; finally, (5) the OSSECO predominant researchers. They are important in order to identify the keystone authors in the growing network of OSSECO researchers. Section 2.7.1 describes the summary of the answers to these questions.

SM-RQ2. What is an OSSECO?

To answer this question, we get relevant information about OSSECO domain, but also about OSS and SECO related concepts. We searched for OSSECO definitions, elements belonging to OSSECOs and OSSECO instances reported in literature. Section 2.7.2 describes the summary of the answers to these questions.

Which presentations have been proposed for OSSECOs?

In order to identify which are the representations proposed in the literature for OSSECO, we identified: modelling techniques, analysis, particular notations and guidelines in the literature. Section 2.7.3 describes the summary of the results in this topic.

> **RQ2:** Is it feasible to use the QuESo-framework to evaluate the quality of OSSECOs?

The design problem of this thesis is twofold, identify the knowledge about of OSSECO as well as to provide the QuESo-framework in order to support the quality evaluation of OSSECOs. QuESo-framework is divided in three artifacts:

- **QuESo-model**, it is a model for the quality assessment of OSSECOs. This model provides three-dimensional perspectives for evaluating OSSECOs: (1) those that relate to the platform around which the ecosystem is built; (2) those that relate to the community (or set of communities) of the ecosystem and (3) those that are related to the ecosystem as a network of interrelated elements, such as projects or companies. QuESo-model is described in detail in Chapter 3.
- **QuESo-process** is designed for the quality evaluation of OSSECOs. We propose an approach based on building Bayesian networks (BN) as evaluation model. These networks are derived systematically from: the QuESo-model, historical data from OSSECOs and empirical survey data from experts. QuESo-process is divided into two sub-processes. (1) Measurement process is a three-step process to assess the available measures of the OSSECO. (2) Evaluation process is a fourth-step process to establish a set of assessment goals from experts and to make the assessment of QuESo-model subcharacteristics. QuESo-process is described in detail in Chapter 4.
- **QuESo-tool** is a set of tools to support semi-automatic quality evaluation of OSSECOs. The tool is a set of a set of software components that support the QuESo-process in order to evaluate the quality of OSSECOs. QuESo-tool is described in detail in Chapter 6.

To answer the **RQ2**, we conducted a case study to evaluate the quality of Eclipse OSSECO. We defined a set of measures (Table 48) in order to answer the research sub questions: (1) What is the validity of the QuESo-quality model? (2) What is the validity of the results obtained applying the QuESo-process? And (3) What is the perceived usefulness of the QuESo-process? In order to answer these questions, we designed three questionnaires for covering the three research questions (see Table 48). (1) First questionnaire was designed to measure the perceived QuESo-model validity (i.e., the RQ2.1). The **RQ2.1** survey consisted of 16 questions (one for each QuESo-model subcharacteristic). Each question was rated on a 5-point Likert scale where 1= *Unimportant* and 5=*Very Important*. (2) Second questionnaire was designed to measure the perceived validity of the results obtained by applying the QuESo-process (i.e., the RQ2.2). Similar to the RQ2.2 survey, The **RQ2.2** survey consisted of 16 questions (related with QuESo-model subcharacteristics). Each question was rated on a 5-point Likert scale where 1=*Strongly Disagree* and 5=*Strongly Agree*. Finally, (3) Third questionnaire was designed to measure the perceived usefulness of the QuESo-process (i.e., the RQ2.3). The **RQ2.3** survey consisted of 5 questions. Each question was rated on a 5-point Likert scale where 1=*Strongly Disagree* and 5=*Strongly Agree.*

For each survey, we defined a set of hypotheses, and we used the procedures defined by Boone and Boone for Likert-type data (i.e., statistical median for *Central tendency*, frequencies for *Variability* and *Wilcoxon signed rank* for hypotheses testing) [185]. The hypothesis testing process shows that, in general, the surveys participants perceive a positive validity of the QuESo-model, a positive validity of the results obtained applying the QuESo-process and a positive usefulness of the QuESo-process. The QuESo-process validation is detailed in Chapter 5.

8 Bibliography

[1] R. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*, 1st ed. Springer, 2014.

[2] M. Shaw, “The coming-of-age of software architecture research,” in *Proceedings of the 23rd international conference on Software engineering*, 2001, p. 656.

[3] L. López *et al.*, “Adoption of OSS components: A goal-oriented approach,” *Data Knowl. Eng.*, vol. 99, pp. 17–38, 2015.

[4] G. M. Kapitsaki, N. D. Tselikas, and I. E. Foukarakis, “An insight into license tools for open source software systems,” *J. Syst. Softw.*, vol. 102, pp. 72–87, 2015.

[5] L. Uden, E. Damiani, G. Gianini, and P. Ceravolo, “Activity theory for OSS ecosystems,” in *Proceedingd of the Digital EcoSystems and Technologies Conference, DEST*, 2007, pp. 223–228.

[6] D. Riehle, “The economic motivation of open source software: Stakeholder perspectives,” *Computer (Long. Beach. Calif).*, vol. 40, no. 4, pp. 25–32, 2007.

[7] B. D. Ripley, “The R project in statistical computing,” *MSOR Connect.*, vol. 1, no. 1, pp. 23–25, 2001.

[8] M. Foulonneau, R. Pawelzik, B. Grégoire, and O. Donak, “Analyzing the Open Source communities’ lifecycle with communication data,” in *Proceedings of the 5th MEDES*, 2013, pp. 340--344.

[9] M. Squire and D. Williams, “Describing the software forge ecosystem,” in *Proceedings of the 45th System Science , Hawaii International Conference*, 2012, pp. 3416–3425.

[10] M. Zhou, A. Mockus, X. Ma, L. Zhang, and H. Mei, “Inflow and retention in oss communities with commercial involvement: A case study of three hybrid projects,” *ACM -(TOSEM)*, vol. 25, no. 2, pp. 13–29, 2016.

[11] N. Figay and P. Ghodous, “FLOSS as Enterprise Application Interoperability Enabler,” in *Proceedings of the 5th Signal-Image Technology & Internet-Based Systems International Conference (SITIS)*, 2009.

[12] M. Goeminne and T. Mens, “Analyzing ecosystems for open source software developer communities,” in *Software Ecosystems Analyzing and Managing Business Networks in the Software Industry*, 1st ed., S. Jansen, S. Brinkkemper, and M. A. Cusumano, Eds. Edward Edgar Publishing Limited, 2013, pp. 247–275.

[13] J. Buford, K. Mahajan, and V. Krishnaswamy, “Federated enterprise and cloud-based collaboration services,” in *Proceedings of the IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application*, 2011, pp. 1–6.

[14] M. Goeminne, “Understanding the Evolution of Socio-Technical Aspects in Open Source Ecosystems,” in *Proccedings of IEEE-(CSMR-WCRE)*, 2014, pp. 473–476.

[15] B. Fitzgerald and P. J. Agerfalk, “Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy,” *MIS Quartely*, vol. 32, no. 2, pp. 385–409, 2008.

[16] T. Mens and M. Goeminne, “Analysing the Evolution of Social Aspects of Open Source Software Ecosystems,” in *Proceedings of the 3rd International Workshop on Software Ecosystems (IWSECO)*, 2011, pp. 1–14.

[17] T. Xie, T. Zimmermann, and A. van Deursen, “Introduction to the Special Issue on Mining Software Repositories.,” *Empir. Softw. Eng.*, vol. 18, no. 6, pp. 1043–1046, 2013.

[18] G. Gousios and D. Spinellis, “GHTorrent: Github’s Data from a Firehose,” in *Proceeding of the 9th MSR*, 2012, pp. 12–21.

[19] D. G. Messerschmitt and C. Szyperski, *Software Ecosystem: Understanding an Indispensable Technology and Industry*, 1\textsupe., vol. 1. Cambridge MA, USA: The MIT Press, 2003.

[20] O. Franco-Bedoya, D. Ameller, D. Costal, and X. Franch, “QuESo: A quality model for open source software ecosystems,” in *ICSOFT-EA 2014 - Proceedings of the 9th International Conference on Software Engineering and Applications*, 2014.

[21] T. Berger *et al.*, “Variability mechanisms in software ecosystems,” *Inf. Softw. Technol.*, vol. 56, no. 11, pp. 1520–1535, 2014.

[22] T. Mens, M. Claes, P. Grosjean, and A. Serebrenik, “Studying Evolving Software Ecosystems based on Ecological Models,” in *Evolving Software Systems*, T. Mens, A. Serebrenik, and A. Cleve, Eds. 2014, pp. 297–326.

[23] J. Bosch, “From software product lines to software ecosystems,” *Proc. 13th Int. Softw. Prod. Line Conf.*, pp. 111–119, 2009.

[24] S. Jansen, A. Finkelstein, and S. Brinkkemper, “A Sense of Community: A Research Agenda for Software Ecosystems,” in *Proc. of the 31st Int’l Conference on Software Engineering - Companion Volume (ICSE ’09)*, 2009, pp. 187–190.

[25] S. Jansen, A. Finkelstein, and A. Finkelstein, “Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems,” in *Proceedings of the 1st IWSECO*, 2009, pp. 187–190.

[26] R. P. dos Santos and C. M. L. Werner, “A Proposal for Software Ecosystems Engineering.,” in *Proceedings of the 3th IWSECO*, 2011, pp. 40–51.

[27] M. Lungu, “Towards reverse engineering software ecosystems,” in *Proceedings of the IEEE International Conference on Software Maintenance*, 2008, pp. 428–431.

[28] M. Goeminne and T. Mens, “A framework for analysing and visualising open source software ecosystems,” in *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE)*, 2010, pp. 42–47.

[29] L. Yu, “The Market-Driven Software Ecosystem,” *IT Prof.*, vol. 15, no. 5, pp. 46–50, Sep. 2013.

[30] U. Eklund and J. Bosch, “Architecture for Embedded Open Software Ecosystems,” *J. Syst. Softw.*, vol. 92, pp. 128–142, 2014.

[31] D. Wynn Jr., “Assessing the Health of an Open Source Ecosystem,” in *Emerging Free and Open Source Software Practices*, S. K. Sowe, I. G. Stamelos, and I. Samoladas, Eds. IGI Global, 2007, pp. 238–258.

[32] S. Jansen and M. A. Cusumano, “Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance,” in *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, Edward Elgar Publishing, 2013, pp. 13--28.

[33] R. Hoving, G. Slot, and S. Jansen, “Python: Characteristics Identification of a Free Open Source Software Ecosystem,” in *Proceddings of the 7th IEEE-(DEST)*, 2013, pp. 13–18.

[34] O. Franco-Bedoya, D. Ameller, D. Costal, and X. Franch, “Open source software ecosystems: A Systematic mapping,” *Inf. Softw. Technol.*, vol. 91, pp. 160–185, Nov. 2017.

[35] W. Scacchi, “Free/Open Source Software Development : Recent Research Results and Emerging Opportunities,” in *Proceedings of the The 6th Joint ACM SIGSOFT Conference*, 2007, pp. 459–468.

[36] A. Wasserman, “Community and Commercial Strategies in Open Source Software,” *Inf. Technol.*, vol. 55, no. 5, pp. 181–188, 2013.

[37] O. Barbosa, R. P. dos Santos, C. Alves, C. Werner, and S. Jansen, “Systematic Mapping Study on Software Ecosystems from a Three-Dimensional Perspective.,” in *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, vol. 1, 2013, pp. 59–81.

[38] G. K. Hanssen and T. Dybå, “Theoretical Foundations of Software Ecosystems,” in *Proceedings of the 4th IWSECO-ICSOB*, 2012, pp. 6–17.

[39] K. Manikas, “Revisiting Software Ecosystems Research: A Longitudinal Literature Study,” *J. Syst. Softw.*, vol. 117, pp. 84–103, 2016.

[40] A. Adewumi, S. Misra, and N. Omoregbe, “A Review of Models for Evaluating Quality in Open Source Software,” in *Proceedings o the IEECS*, 2013, vol. 4, pp. 88–92.

[41] F.-W. Duijnhouwer and C. Windows, “Open Source maturity model:,” *Cap Gemini Expert Lett.*, vol. 2009, no. June 6, 2003.

[42] R. Semeteys, “Qualification and Selection of Open Source Software (QSOS),” *Open Source Bus. Resour.*, no. May 2008, pp. 1–29, 2013.

[43] A. Wasserman, M. Pal, and C. Chan, “Business Readiness Rating for Open Source,” in *Proceedings of the EFOSS*, 2006.

[44] W. J. Sung, J. H. Kim, and S. Y. Rhew, “A Quality Model for Open Source Software Selection,” in *Proceedings of 6th ALPIT*, 2007, pp. 515–519.

[45] M. Soto and M. Ciolkowski, “The QualOSS open source assessment model measuring the performance of open source communities,” in *2009 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2009, pp. 498–501.

[46] E. Petrinja, R. Nambakam, and A. Sillitti, “Introducing the OpenSource Maturity Model,” in *Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*, 2009, pp. 37–41.

[47] I. Samoladas, G. Gousios, D. Spinellis, and I. Stamelos, “The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation,” in *Proceddings of 20th Open Source Development, Communities and Quality: IFIP 20th World Computer Congress*, 2008, pp. 237–248.

[48] L. Aversano and M. Tortorella, “Quality evaluation of floss projects: Application to ERP systems,” *Inf. Softw. Technol.*, vol. 55, no. 7, pp. 1260–1276, 2013.

[49] J. Axelsson and M. Skoglund, “Quality Assurance in Software Ecosystems: A Systematic Literature Mapping and Research Agenda,” *J. Syst. Softw.*, vol. 114, pp. 69–81, 2016.

[50] O. Franco-Bedoya, D. Ameller, D. Costal, and X. Franch, “QuESo: A Quality Model for Open Source Software Ecosystems,” in *Proceedings of the 9th International Conference on Software Engineering and Applications ICSOFT-EA 2014*, 2014, pp. 209–218.

[51] N. Dai *et al.*, “OSAMI Commons - An open dynamic services platform for ambient intelligence,” in *Proceedings of the 16th ETFA*, 2011, pp. 1–10.

[52] T. Kilamo, I. Hammouda, T. Mikkonen, and T. Aaltonen, “From Proprietary to Open Source-Growing an Open Source Ecosystem,” *J. Syst. Softw.*, vol. 85, no. 7, pp. 1467–1478, 2012.

[53] M. Oriol, X. Franch, J. Marco, and D. Ameller, “Monitoring Adaptable SOA-Systems using SALMon,” in *Proceedins of Workshop on Service Monitoring, Adaptation and Beyond (Mona+)*, 2008, pp. 19–28.

[54] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic Mapping Studies in Software Engineering,” in *Proceedings of the 12th EASE*, 2008, pp. 68–77.

[55] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update,” *Inf. Softw. Technol.*, vol. 64, pp. 1–18, 2015.

[56] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3,” 2007.

[57] O. Franco-Bedoya, “Open Source Software Ecosystems: Towards a Modelling Framework,” in *Proceddings of the 11th IFIP*, 2015, pp. 171–179.

[58] O. Franco-Bedoya, D. Ameller, D. Costal, and X. Franch, “Measuring the quality of open source software ecosystems using QuESo,” in *Proceedings of the International Conference on Software Technologies*, 2015, vol. 555, pp. 39–62.

[59] OMG, “Software & Systems Process Engineering Metamodel™ Specification (SPEM™),” 2008. .

[60] M. Oriol, O. Franco-Bedoya, X. Franch, and J. Marco, “Assessing Open Source Communities’ Health Using Service Oriented Computing Concepts,” in *Proceedings of RCIS*, 2014.

[61] O. Franco-Bedoya *et al.*, “Towards Assessing Open Source Communities’ Health Using SOC Concepts,” in *Actas (JCIS 2015)*, 2015, pp. 1–3.

[62] S. Jansen, “Measuring the Health of Open Source Software Ecosystems: Beyond the Scope of Project Health,” *Inf. Softw. Technol.*, vol. 56, no. 11, pp. 1508–1519, 2014.

[63] M. Oriol, J. Marco, and X. Franch, “Quality models for web services: A systematic mapping,” *Inf. Softw. Technol.*, vol. 56, no. 10, pp. 1167–1182, 2014.

[64] O. Barbosa and C. Alves, “A Systematic Mapping Study on Software Ecosystems,” in *Proceedings of the Workshop on Software Ecosystems 2011*, 2011, pp. 15–26.

[65] K. Manikas and K. M. Hansen, “Software ecosystems – A systematic literature review,” *J. Syst. Softw.*, vol. 86, pp. 1294–1306, 2013.

[66] K. Manikas, “Supporting the Evolution of Research in Software Ecosystems: Reviewing the Empirical Literature,” in *Proceedings of the 7th Software Business International Conference, ICSOB 2016*, A. Maglyas and A.-L. Lamprecht, Eds. Cham: Springer International Publishing, 2016, pp. 63–78.

[67] Ø. Hauge, C. Ayala, and R. Conradi, “Adoption of Open Source Software in Software-intensive Organizations--A Systematic Literature Review,” *Inf. Softw. Technol.*, vol. 52, no. 11, pp. 1133–1154, 2010.

[68] *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry [Hardcover]*. Edward Elgar Pub, 2013.

[69] T. Mens, A. Serebrenik, and A. Cleve, *Evolving Software Systems*, 1st ed., vol. 1. Beerlin,Germany: Springer, 2014.

[70] D. Ameller, X. Burgués, O. Collell, D. Costal, X. Franch, and M. P. Papazoglou, "Development of Service-Oriented Architectures Using Model-Driven Development: A Mapping Study ," *Inf. Softw. Technol.*, vol. 62, pp. 42–66, 2015.

[71] Ø. Hauge, C. Ayala, and R. Conradi, "Adoption of Open Source Software in Software-intensive Organizations--A Systematic Literature Review," *Inf. Softw. Technol.*, vol. 52, no. 11, pp. 1133–1154, 2010.

[72] S. Jansen, "Customer Configuration Updating in a Software Supply Network," Utrecht University, 2007.

[73] D. G. Messerschmitt and C. Szyperski, *Software Ecosystem: Understanding an Indispensable Technology and Industry*, 1st ed., vol. 1. Cambridge MA,USA: The MIT Press, 2003.

[74] L. Morgan, J. Feller, and P. Finnegan, "Exploring value networks," *Eur. J. Inf. Syst.*, vol. 22, no. 5, pp. 569–588, 2013.

[75] M. B. Miles, A. M. Huberman, and S. Johnny, *Qualitative Data Analysis: A Methods Sourcebook*, 3rd ed. California, USA: SAGE Publications, 2014.

[76] L. Ben Jabeur, L. Tamine, and M. Boughanem, "A Social Model for Literature Access: Towards a Weighted Social Network of Authors," in *Proceedings of 9th RIAO*, 2010, pp. 32–39.

[77] T. Toivanen, O. Mazhelis, and E. Luoma, "Network Analysis of Platform Ecosystems: The Case of Internet of Things Ecosystem," *Proceedings of the 6th International Conference of Software Business*. 2015.

[78] Y. Kamei, S. Matsumoto, H. Maeshima, Y. Onishi, M. Ohira, and K. Matsumoto, "Analysis of Coordination Between Developers and Users in the Apache Community," in *Proceedings of the 4th OSS*, 2008, pp. 81–92.

[79] J. Golbeck, *Analyzing the Social Web*, 1st ed. Waltham, USA: Elsevier, 2013.

[80] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web.," 1999.

[81] M. E. J. Newman, "Scientific Collaboration Networks. II. Shortest Paths, Weighted Networks, and Centrality," *Phys. Rev. E*, vol. 64, no. 1, p. 16132, Jun. 2001.

[82] M. Montesi and P. Lago, "Software Engineering Article Types: An Analysis of the Literature," *J. Syst. Softw.*, vol. 81, no. 10, pp. 1694–1714, 2008.

[83] S. Jansen, E. Handoyo, and C. Alves, "Scientists' Needs in Modelling Software Ecosystems," in *Proceedings of the ECSAW*, 2015, p. 44:1--44:6.

[84] S. Syed and S. Jansen, "On Clusters in Open Source Ecosystems.," in *Proceedings of the 5th IWSECO*, 2013, pp. 19–32.

[85] R. Spauwen and S. Jansen, "Towards the Roles and Motives of Open Source Software Developers," in *Proceedings of 5th IWSECO*, 2013, p. 62.

[86] E. Ververs, R. van Bommel, and S. Jansen, "Influences on developer participation in the Debian software ecosystem," in *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, 2011, pp. 89–93.

[87] S. Van Lingen, A. Palomba, and G. Lucassen, "On the Software Ecosystem Health of Open Source Content Management Systems," in *Proceedings of the 5th IWSECO*, 2013, p. 38.

[88] J. Bosch, "Software Ecosystems - Implications for Strategy, Business Model and Architecture," in *Proceedings of the 15th Software Product Line Conference (SPLC)*, 2012.

[89] SCIMago, “Journal & Country Rank SCImago. (2007).” Sep-2015.

[90] W. Li, X. Wang, and M. Yu, “A Research on Collaboration Knowledge Construction in the Virtual Learning Community by Social Network Analysis,” in *Proceedings of ICEIT*, 2010, vol. 2, pp. V2-323-V2-327.

[91] A. Abbasi and J. Altmann, “On the Correlation Between Research Performance and Social Network Analysis Measures Applied to Research Collaboration Networks,” in *Proceedings of the 44th HICSS*, 2011, pp. 1–10.

[92] X. Franch and J. P. Carvallo, “Using quality models in software package selection,” *IEEE Softw.*, vol. 20, pp. 34–41, 2003.

[93] F. Cheong and B. Corbitt, “A Social Network Analysis of the Co-authorship Network of the Pacific Asia Conference on Information Systems from 1993 to 2008,” in *Proceedings of PACIS 2009*, 2009, pp. 1–23.

[94] F. Ruíz, “La Investigación en Informática. en España Análisis Bibliométrico,” *Novática*, vol. 85, no. 215, pp. 54–58, 2012.

[95] W. Chen and R. Hirschheim, “A Paradigmatic and Methodological Examination of Information Systems Research From 1991 to 2001,” *Inf. Syst. J.*, vol. 14, no. 3, pp. 197–235, 2004.

[96] V. Boucharas, S. Jansen, and S. Brinkkemper, “Formalizing software ecosystem modeling,” *Proc. 1st Int. Work. Open Compon. Ecosyst. IWOCE 09*, vol. 19, p. 41, 2009.

[97] G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval ,” *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988.

[98] I. Feinerer, “Introduction to the tm Package Text Mining in R,” 2015. [Online]. Available: https://cran.r-project.org/web/packages/tm/index.html.

[99] T. Narang, “Hiearchical Clustering of Documents: A Brief Study and Implementation in Matlab,” in *Proceedings of the 3rd ICETESMA*, 2015, pp. 153–159.

[100] M. Scott and C. Tribble, *Textual Patterns: Key Words and Corpus Analysis in Language Education*, 1st ed., vol. 22. Amsterdam, The Netherlands: John Benjamins Publishing, 2006.

[101] P. Rayson, D. Berridge, and B. Francis, “Extending the Cochran Rule for the Comparison of Word Frequencies Between Corpora,” in *Proceedings of 7th*, 2004, pp. 926–936.

[102] M. Davies, “The corpus of contemporary American English,” 2015. [Online]. Available: http://www.academicvocabulary.info/.

[103] E. Taljard and G.-M. De Schryver, “Semi-automatic Term Extraction for the African Languages, with Special Reference to Northern Sotho,” *Lexikos*, vol. 12, no. 1, 2002.

[104] D. M. German, B. Adams, and A. E. Hassan, “The Evolution of the R Software Ecosystem,” in *Proceddings of the 17th CSMR*, 2013, pp. 243–252.

[105] J. van Angeren, J. Kabbedijk, S. Jansen, and K. Popp, “A survey of associate models used within large software ecosystems,” in *Proceedings of the third Workshop on Software Ecosystems, Brussels, Belgium, CEUR-WS*, 2011, pp. 27–39.

[106] E. Handoyo, S. Jansen, and S. Brinkkemper, “Software Ecosystem Modeling: The Value Chains,” in *Proceedings of the 5th MEDES*, 2013, pp. 17–24.

[107] E. Yu and D. Stephanie, “Understanding Software Ecosystems: A Strategic Modeling Approach,” in *Proceedings of the 3rd IWSECO*, 2011, pp. 65–76.

[108] D. Wynn Jr., M.-C. Boudreau, and R. T. Watson, "Resilience of professional open source ecosystems," in *ECIS 2008 PROCEEDINGS*, 2008, p. Paper 102.

[109] M. H. Sadi and E. Yu, "Designing Software Ecosystems: How Can Modeling Techniques Help?," in *Proceedings of 16th BPMDS*, 2015, pp. 360–375.

[110] H. B. Christensen, K. M. Hansen, M. Kyng, and K. Manikas, "Analysis and Design of Software Ecosystem Architectures : Towards the 4S Telemedicine Ecosystem," *Inf. Softw. Technol.*, vol. 56, no. 11, pp. 1476–1492, 2014.

[111] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*, 1st ed. New York, USA: Springer Science & Business Media, 2012.

[112] S. Jansen, S. Brinkkemper, and Michael Cusumano, Eds., *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry (Google eBook)*, 1st ed. London: Edward Elgar Publishing, 2013.

[113] "ISO/IEC 25000:2005 - Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE," 2014. [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=35683. [Accessed: 24-Mar-2014].

[114] E. den Hartigh, W. Visscher, M. Tol, and A. J. Salas, "Measuring the health of a business ecosystem," in *Software Ecosystems Analyzing and Managing Business Networks in the Software Industry*, 1st ed., S. Jansen, S. Brinkkemper, and M. Cusumano, Eds. Edward Edgar Publishing Limited, 2013, pp. 221–246.

[115] M. Iansiti and R. Levien, "Keystones and dominators: framing operating and technology strategy in a business ecosystem," *Harvard Bus. Sch. Bost.*, 2004.

[116] S. Neu, M. Lanza, L. Hattori, and M. D'Ambros, "Telling stories about GNOME with Complicity," in *Visualizing Software for Understanding and Analysis (VISSOFT), 2011 6th IEEE International Workshop on*, 2011, pp. 1–8.

[117] T. Kilamo, I. Hammouda, T. Mikkonen, and T. Aaltonen, "From proprietary to open source—Growing an open source ecosystem," *J. Syst. Softw.*, vol. 85, no. 7, pp. 1467–1478, Jul. 2012.

[118] M. Kläs, C. Lampasona, S. Nunnenmacher, S. Wagner, M. Herrmannsdörfer, and K. Lochmann, "How to evaluate meta-models for software quality?," in *Proceedings of IWSM - MetriKon*, 2010, pp. 443--462.

[119] S. Wagner, "A Bayesian network approach to assess and predict software quality using activity-based quality models," *Inf. Softw. Technol.*, vol. 52, no. 11, pp. 1230–1241, 2010.

[120] A. Adewumi, S. Misra, and N. Omoregbe, "Evaluating Open Source Software Quality Models Against ISO 25010," in *Proceedings of the IEEE (CIT/IUCC/DASC/PICOM)*, 2015, pp. 872–877.

[121] X. Franch and J. P. Carvallo, "Using quality models in software package selection," *IEEE Softw.*, vol. 20, pp. 34–41, 2003.

[122] B. Behkamal, M. Kahani, and M. K. Akbari, "Customizing ISO 9126 quality model for evaluation of B2B applications," *Inf. Softw. Technol.*, vol. 51, pp. 599–609, 2009.

[123] F. Radulovic and R. Garcia-Casiro, "Extending software quality models - A sample in the domain of semantic technologies," in *SEKE 2011 - Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering*, 2011, pp. 25–30.

[124] S. Jansen and M. Cusumano, "Defining software ecosystems: a survey of software platforms and business network governance : Software Ecosystems Analyzing and Managing Business Networks in the Software Industry," in *Software Ecosystems Analyzing and Managing*

Business Networks in the Software Industry, 1st ed., S. Jansen, S. Brinkkemper, and Michael Cusumano, Eds. London, UK: Edward Edgar Publishing Limited, 2013, pp. 13–28.

[125] O. Franco-Bedoya, D. Ameller, D. Costal, and X. Franch, "QuESo V2. 0 a Quality Model for Open Source Software Ecosystems: List of Measures," 2016.

[126] J. C. Anderson, J. A. Narus, and D. Narayandas, *Business Market Management: Understanding, Creating, and Delivering Value (3rd Edition)*. Prentice Hall, 2009.

[127] X. Li, X. Jie, Q. Li, and Q. Zhang, "Research on the Evaluation of Business Ecosystem Health," in *Proceedings of the Sixth International Conference on Management Science and Engineering Management*, 2013, pp. 1009–1020.

[128] J. Gamalielsson, B. Lundell, and B. Lings, "The Nagios community: An extended quantitative analysis," in *IFIP Advances in Information and Communication Technology*, 2010, vol. 319 AICT, pp. 85–96.

[129] B. Lundell and B. Forssten, "Exploring health within OSS ecosystems," in *Proceedings of the First International Workshop on Building Sustainable Open Source Communities*, 2009, pp. 1–5.

[130] F. van der Linden, B. Lundell, and P. Marttiin, "Commodification of Industrial Software: A Case for Open Source," *IEEE Softw.*, vol. 26, 2009.

[131] W. Scacchi and T. A. Alspaugh, "Understanding the role of licenses and evolution in open architecture software ecosystems," *J. Syst. Softw.*, vol. 85, pp. 1479–1494, 2012.

[132] M. Lungu, M. Lanza, T. Gîrba, and R. Robbes, "The small project observatory: Visualizing software ecosystems," *Sci. Comput. Program.*, vol. 75, no. 4, pp. 264–275, 2010.

[133] H. Haken, "Synergetics Are Cooperative Phenomena Governed by Universal Principles ?," *Naturwissenschaften*, vol. 67, pp. 121–128, 1980.

[134] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*, vol. 2640. 1992.

[135] R. Glott, K. Haaland, and S. Bannier, "D3.1 Draft Business Model Risk Requirements Report." Barcelona, 2013.

[136] M. Goeminne and T. Mens, "Analyzing ecosystems for open source software developer communities," in *Software Ecosystems Analyzing and Managing Business Networks in the Software Industry*, 1st ed., S. Jansen, S. Brinkkemper, and M. Cusumano, Eds. Edward Edgar Publishing Limited, 2013, pp. 247–275.

[137] C. Jergensen and A. Sarma, "The onion patch: migration in open source ecosystems," *Proc. 19th ACM SIGSOFT Symp. 13th Eur. Conf. Found. Softw. Eng.*, pp. 70–80, 2011.

[138] S. Lohmann, S. Negru, F. Haag, and T. Ertl, "Visualizing ontologies with VOWL," *Semant. Web*, vol. 7, no. 4, pp. 399–419, 2016.

[139] D. Dhungana, I. Groher, E. Schludermann, and S. Biffl, "Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature," in *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, 2010, pp. 96–102.

[140] R. E. Rolfe, "Social cohesion and community resilience: A multi-disciplinary review of literature for rural health research," *Halifax Dep. Int. Dev. Stud. Fac. Grad. Stud. Res. Saint Mary's Univ.*, 2006.

[141] S. A. Hissam, B. Russo, M. G. Mendonça Neto, and F. Kon, Eds., *Open Source Systems: Grounding Research*, vol. 365. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[142] K. Manikas and K. M. Hansen, “Reviewing the health of software ecosystems--a conceptual framework proposal,” in *Proceedings of the 5th International Workshop on Software Ecosystems (IWSECO)*, 2013, pp. 33–44.

[143] S. Jansen, “Measuring the Health of Open Source Software Ecosystems: Moving Beyond the Project Scope,” *Inf. Softw. Technol.*, p. in Press, 2014.

[144] S. da Silva Amorim, E. S. de Almeida, J. D. McGregor, and C. von Flach G. Chavez, “Towards an Evaluation Method for Software Ecosystem Practices,” in *Proccedings of the 10th European Conference on Software Architecture Workshops*, 2016, p. 25:1--25:4.

[145] T. Mens, L. Doctors, N. Habra, B. Vanderose, and F. Kamseu, “QUALGEN: Modeling and Analysing the Quality of Evolving Software Systems,” *Proc. 15th Eur. Conf. Softw. Maint. Reengineering*, pp. 351–354, 2011.

[146] F. Fotrousi, S. A. Fricker, M. Fiedler, and F. Le-Gall, “KPIs for Software Ecosystems: A Systematic Mapping Study,” in *Software Business. Towards Continuous Value Delivery: 5th International Conference, ICSOB 2014, Paphos, Cyprus, June 16-18, 2014. Proceedings*, C. Lassenius and K. Smolander, Eds. Cham: Springer International Publishing, 2014, pp. 194–211.

[147] N. Fenton and J. Bieman, *Software metrics: a rigorous and practical approach*. CRC Press, 2014.

[148] J. M. Beaver, G. A. Schiavone, and J. S. Berrios, “Predicting software suitability using a Bayesian belief network,” in *Proceedings of 4th International Conference on Machine Learning and Applications (ICMLA’05)*, 2005, p. 7 pp.-pp.

[149] A. L. Madsen, M. Lang, U. B. Kjærulff, and F. Jensen, “The Hugin tool for learning Bayesian networks,” in *Proceedings of ECSQARU*, editors T.D. Nielsen and N.L. Zhang, Ed. Berlin Heidelberg: Springer-Verlag, 2003, pp. 594–605.

[150] ISO/IEC-25010, *Systems and Software Engineering—Systems and Software Product Quality Requirements and Evaluation (SQuaRE)— System and Software Quality Models*. Geneva, 2010.

[151] B. Kitchenham, S. Linkman, A. Pasquini, and V. Nanni, “The SQUID approach to defining a quality model,” *Softw. Qual. J.*, vol. 6, no. 3, pp. 211–233, 1997.

[152] F. Radulovic, R. García-Castro, and A. Gómez-Pérez, “SemQuaRE—An extension of the SQuaRE quality model for the evaluation of semantic technologies,” *Comput. Stand. Interfaces*, vol. 38, pp. 101–112, 2015.

[153] B. W. Boehm, J. R. Brown, and M. Lipow, “Quantitative Evaluation of Software Quality,” in *Proceedings of the 2nd ICSE*, 1976, pp. 592–605.

[154] G. Malak, H. Sahraoui, L. Badri, and M. Badri, “Modeling Web Quality Using a Probabilistic Approach: An Empirical Validation,” *ACM Trans. Web*, vol. 4, no. 3, p. 9:1--9:31, Jul. 2010.

[155] A. Trendowicz, M. Kläs, C. Lampasona, J. Münch, C. Körner, and M. Saft, “Model-based Product Quality Evaluation with Multi-Criteria Decision Analysis,” *CoRR*, vol. abs/1401.1, 2014.

[156] F. Cugnata, R. Kenett, and S. Salini, “Bayesian Network Applications to Customer Surveys and InfoQ,” *Procedia Econ. Financ.*, vol. 17, pp. 3–9, 2014.

[157] Z. Sun and D. Müller, “A framework for modeling payments for ecosystem services with agent-based models, Bayesian belief networks and opinion dynamics models,” *Environ. Model. Softw.*, vol. 45, pp. 15–28, 2013.

[158] P. Judea, *Causality: Models, Reasoning and Inference*, 2nd ed. Los Angeles, CA, USA: Cambridge University Press., 2009.

[159] N. Fenton and M. Neil, "Managing Risk in the Modern World," *Appl. Bayesian Networks*, 2007.

[160] R. S. Kenett, "Applications of bayesian networks," *SSRN*, 2012.

[161] T. Koski and J. Noble, *Bayesian networks: an introduction*, vol. 924. John Wiley & Sons, 2011.

[162] M. NEIL, N. FENTON, and L. NIELSON, "Building large-scale Bayesian networks," *Knowl. Eng. Rev.*, vol. 15, no. 3, pp. 257–284, 2000.

[163] A. C. Constantinou, N. Fenton, and M. Neil, "Integrating expert knowledge with data in Bayesian networks: Preserving data-driven expectations when the expert variables remain unobserved," *Expert Syst. Appl.*, vol. 56, pp. 197–208, 2016.

[164] R. W. Ferguson, D. Goldenson, J. M. McCurley, R. W. Stoddard, and D. Zubrow, "Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE)," in *SEI Technical report*, 2011, no. December, pp. 1–80.

[165] S. L. Lauritzen, "The EM algorithm for graphical association models with missing data," *Comput. Stat. Data Anal.*, vol. 19, no. 2, pp. 191–201, 1995.

[166] L. M. Campos, "A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests," *J. Mach. Learn. Res.*, vol. 7, no. 2, pp. 2149--2187, 2007.

[167] J. A. Wellner and Y. Zhan, "A Hybrid Algorithm for Computation of the Nonparametric Maximum Likelihood Estimator from Censored Data," *J. Am. Stat. Assoc.*, vol. 92, no. 439, pp. 945–959, 1997.

[168] Y. Zhou, N. Fenton, M. Neil, and C. Zhu, "Incorporating Expert Judgement into Bayesian Network Machine Learning," in *Proceedings of the 23th IJCAI*, 2013, pp. 3249–3250.

[169] A. Zagorecki and M. J. Druzdzel, "Probabilistic Independence of Causal Influences.," in *Proceeding of the third PGM, Prague, Czech Republic*, 2006, pp. 325–332.

[170] N. E. Fenton and M. Neil, "A critique of software defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 25, no. 5, pp. 675–689, Sep. 1999.

[171] X. Franch *et al.*, "The RISCOSS Platform for Risk Management in Open Source Software Adoption," in *Proceedings of IFIP OSS Conference*, 2015, pp. 124–133.

[172] O. Z. Khan, P. Poupart, and J. M. Agosta, "Automated Refinement of Bayes Networks\textquotesingle Parameters based on Test Ordering Constraints," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2591–2599.

[173] M. Ruiz, D. Costal, S. España, X. Franch, and Ó. Pastor, "GoBIS: An Integrated Framework to Analyse the Goal and Business Process Perspectives in Information Systems," *Inf. Syst.*, vol. 53, pp. 330–345, 2015.

[174] C. G. von Wangenheim, J. C. R. Hauck, A. Zoucas, C. F. Salviano, F. McCaffery, and F. Shull, "Creating Software Process Capability/Maturity Models," *IEEE Softw.*, vol. 27, no. 4, pp. 92–94, Jul. 2010.

[175] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering: Guidelines and Examples*. John Wiley & Sons, 2012.

[176] R. K. Yin, *Case Study Research, Design and Methods*. SAGE, 2014.

[177] P. Brereton, B. Kitchenham, D. Budgen, and Z. Li, “Using a Protocol Template for Case Study Planning,” in *Proceedings of the 12th IEASE*, 2008.

[178] V. R. Basili and D. M. Weiss, “A Methodology for Collecting Valid Software Engineering Data,” *IEEE Trans. Softw. Eng.*, vol. 10, no. 6, pp. 728–738, Nov. 1984.

[179] V. R. Basili *et al.*, “Determining the Impact of Business Strategies Using Principles from Goal-oriented Measurement.,” in *Proceedings of the 9th WI*, 2009, pp. 545–554.

[180] G. Stevens and S. Draxler, “Appropriation of the Eclipse Ecosystem: Local Integration of Global Network Production,” in *Proc. of COOP*, 2010.

[181] K. Mizushima and Y. Ikawa, “A structure of co-creation in an open source software ecosystem: A case study of the eclipse community,” in *2011 Proceedings of PICMET ’11: Technology Management in the Energy Smart World (PICMET)*, 2011, pp. 1–8.

[182] Eclipse, “Eclipse Development Process 2015,” 2015. [Online]. Available: https://www.eclipse.org/projects/dev_process/development_process.php.

[183] 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.

[184] K. B. Wright, “Researching Internet-based populations: Advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services,” *J. Comput. Commun.*, vol. 10, no. 3, p. 0, 2005.

[185] H. N. Boone and D. A. Boone, “Analyzing likert data,” *J. Ext.*, vol. 50, no. 2, pp. 1–5, 2012.

[186] S. Jansen, J. Souer, L. Luinenburg, and S. Brinkkemper, “Shades of gray: Opening up a software producing organization with the open software enterprise model,” *J. Syst. Softw.*, vol. 85, pp. 1495–1510, 2012.

[187] F. D. Davis, “Perceived usefulness, perceived ease of use, and user acceptance of information technology,” *MIS Q.*, pp. 319–340, 1989.

[188] J. Kabbedijk and S. Jansen, “Steering insight: An exploration of the ruby software ecosystem,” in *Proceedings of 2nd International Conference, ICSOB*, 2011, pp. 44–55.

[189] M. Grechanik *et al.*, “An Empirical Investigation into a Large-scale Java Open Source Code Repository,” in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 2010, p. 11:1--11:10.

[190] M. Palacios, J. Garcia-Fanjul, J. Tuya, and G. Spanoudakis, “Identifying Test Requirements by Analyzing SLA Guarantee Terms,” in *Proceedings of the 19th Web Services (ICWS), International Conference*, 2012, pp. 351–358.

[191] M. Comuzzi, C. Kotsokalis, G. Spanoudakis, and R. Yahyapour, “Establishing and Monitoring SLAs in Complex Service Based Systems,” in *Proceedings of the Web Services, ICWS 2009. IEEE International Conference*, 2009, pp. 783–790.

[192] C. Muller *et al.*, “Comprehensive Explanation of SLA Violations at Runtime,” *Serv. Comput. IEEE Trans.*, vol. PP, no. 99, p. 1, 2013.

[193] G. Ghezzi and H. Gall, “A framework for semi-automated software evolution analysis composition,” *Autom. Softw. Eng.*, vol. 20, no. 3, pp. 463–496, 2013.

[194] Y. Raanan, R. S. Kenett, and R. Pike, “Operational risk management: an overview,” in *Operational Risk Management*, First ed., Ron S. Kenett and Yossi Raanan (editors), Ed. Wiley Online Library, 2010, pp. 19–38.

[195] N. Bergida *et al.*, "D2.2 - Intermediate Proposal for Risk Management Techniques," *Teccnical Report. RISCOSS FP7 Project*, 2013.

[196] M. Smith, "{NodeXL} {Network Overview, Discovery and Exploration for Excel}." 2014.

[197] X. Franch *et al.*, "A Layered Approach to Managing Risks in OSS Projects," in *Proceedings of the 10th International Conference on Open Source Systems*, 2014.

[198] O. Cabrera *et al.*, "WeSSQoS: A configurable SOA system for quality-aware web service selection," *arXiv Prepr. arXiv1110.5574*, 2011.

[199] L. Baresi and S. Guinea, "Self-Supervising BPEL Processes," *Softw. Eng. IEEE Trans.*, vol. 37, no. 2, pp. 247–263, 2011.

[200] O. Sammodi, A. Metzger, X. Franch, M. Oriol, J. Marco, and K. Pohl, "Usage-Based Online Testing for Proactive Adaptation of Service-Based Applications," in *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*, 2011, pp. 582–587.

[201] A. Kertesz *et al.*, "Enhancing Federated Cloud Management with an Integrated Service Monitoring Approach," *J. Grid Comput.*, vol. 11, no. 4, pp. 699–720, 2013.

[202] BayesFusion, "jSMILE," 2015. [Online]. Available: https://www.bayesfusion.com/.

[203] A. Bucchiarone, C. Cappiello, E. Di Nitto, R. Kazhamiakin, V. Mazza, and M. Pistore, "Design for Adaptation of Service-Based Applications: Main Issues and Requirements," in *Proceedings of Service-Oriented Computing. ICSOC/ServiceWave Workshops: International Workshops, ICSOC/ServiceWave 2009*, A. Dan, F. Gittler, and F. Toumani, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 467–476.

[204] T. Yamakami, "A three-layer view model of OSS: Toward understanding of diversity of OSS," in *Proceedings of 13th Advanced Communication Technology (ICACT)*, 2011, pp. 1190–1194.

[205] O. Franco-Bedoya, D. Costal, S. Hidalgo, and R. Ben-Jacob, "Expert Mining for Evaluating Risk Indicators Scenarios," in *Proceedings of the (COMPSACW), in the IEEE 38h*, 2014, pp. 205–210.

[206] R. K. Bergida, Nili, Blumenfeld, Yehuda, Ron Ben Jacob, "D2 .2 - Intermediate Proposal for Risk Management Techniques," *RISCOSS project*, vol. 1. RISCOSS Consortium, Barcelona, pp. 1–113, 2013.

[207] O. Franco-Bedoya, D. Ameller, D. Costal, and X. Franch, "Measuring the Quality of Open Source Software Ecosystems Using QuESo," in *Software Technologies: 9th International Joint Conference, ICSOFT 2014, Vienna, Austria, August 29-31, 2014, Revised Selected Papers*, A. Holzinger, J. Cardoso, J. Cordeiro, T. Libourel, L. A. Maciaszek, and M. van Sinderen, Eds. Cham: Springer International Publishing, 2015, pp. 39–62.

[208] O. Franco-Bedoya, *Open source software ecosystems: Towards a modelling framework*, vol. 451. 2015.

[209] O. Franco-Bedoya, D. Ameller, D. Costal, and X. Franch, "Open source software ecosystems: A Systematic mapping," *Inf. Softw. Technol.*, 2017.

[210] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009.

[211] T. A. Alspaugh, H. U. Asuncion, and W. Scacchi, "The role of software licenses in open architecture ecosystems," in *Proceedings of First International Workshop on Software Ecosystems (IWSECO-2009)*, 2009, pp. 4–18.