The Journal of Systems and Software 86 (2013) 1294–1306

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Software ecosystems – A systematic literature review

Konstantinos Manikas*, Klaus Marius Hansen

Department of Computer Science (DIKU), University of Copenhagen, Denmark

ARTICLE INFO

Article history:
Received 28 March 2012
Received in revised form 8 December 2012
Accepted 8 December 2012
Available online 20 December 2012

Keywords:
Software ecosystems
Software ecosystem
Systematic literature review

ABSTRACT

A software ecosystem is the interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services. Arguably, software ecosystems are gaining importance with the advent of, e.g., the Google Android, Apache, and Salesforce.com ecosystems. However, there exists no systematic overview of the research done on software ecosystems from a software engineering perspective. We performed a systematic literature review of software ecosystem research, analyzing 90 papers on the subject taken from a gross collection of 420. Our main conclusions are that while research on software ecosystems is increasing (a) there is little consensus on what constitutes a software ecosystem, (b) few analytical models of software ecosystems exist, and (c) little research is done in the context of real-world ecosystems. This work provides an overview of the field, while identifying areas for future research.

1. Introduction

It has recently been suggested that *software ecosystems* (SECOs) are an effective way to construct large software systems on top of a software platform by composing components developed by actors both internal and external (Bosch, 2009; te Molder et al., 2011). In this setting, software engineering is spread outside the traditional borders of software companies to a group of companies, private persons, or other legal entities.

This differs from traditional outsourcing techniques in that the initiating actor does not necessarily own the software produced by contributing actors and does not hire the contributing actors. All actors, however, coexist in an interdependent way, an example being the iOS ecosystem in which Apple provides review of and a platform for selling applications in return for a yearly fee and 30% of revenues of application sale.[1] This is a parallel to natural ecosystems where the different members of the ecosystems (e.g., the plants, animals, or insects) are part of a food network where the existence of one species depends on the rest.

In addition to iOS, Google's Android ecosystem is a prominent example of a (smartphone) software ecosystem. Such ecosystems are arguably gaining importance commercially: it is, e.g., estimated that in 2012, more smartphones than personal computers will be sold.[2]

While software ecosystems are thus arguably gaining importance, research in software ecosystems is in its infancy, starting in 2005 with Messerschmitt and Szyperski (2005) and now with a dedicated workshop in its third year.[3] Our own literature search (see Section 3) revealed a gross list of 420 published papers on software ecosystems. However, until now there has been no systematic literature review (SLR) of the research literature on software ecosystems, leading to potential issues in identifying research gaps and contributions.

In the context of this, we have conducted a systematic literature review in the field of software ecosystems using the approach of Kitchenham and Charters (2007). As such, the purpose of this literature review is to provide an overview of the research reported in the field and identify possible issues that existing literature is not addressing adequately. This work is intended to function as a snapshot of the research in the field by (i) identifying and analyzing the different definitions of SECOs, (ii) analyzing the growth in research reported per year, (iii) classifying the research by type of result, (iv) defining and analyzing the software architecture and structure of SECOs, and (v) analyzing to which extent research is connected to SECO industry.

1.1. Article structure

The rest of this article is organized as following: in Section 2 we specify the review protocol, in Section 3 we document the extraction of the literature, in Section 4 we analyze the literature and answer the research questions, in Section 5 we list possible threats

* Corresponding author. Tel: +45 23839917.
E-mail addresses: kmanikas@diku.dk (K. Manikas), klausmh@diku.dk (K.M. Hansen).

[1] http://developer.apple.com/programs/ios/distribute.html.
[2] http://www.slideshare.net/CMSummit/ms-internet-trends060710final.
[3] http://www.softwareecosystems.org/workshop/.

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.12.026

to the validity of this work and identify areas not covered from the literature and in Section 6 we conclude.

2. Review protocol

The applied review protocol is based on the guidelines of Kitchenham and Charters (2007). The establishment of the review protocol is necessary to ensure that the literature review is systematic and to minimize researcher bias. As such, the literature review is focused on a set of research questions that serve the aim of this work and derive from the reasons that initiated this review. The review protocol is organized in a way that the research questions define the main areas this study is focusing on. Section 2.2 defines the paper literature extraction strategy including the list of resource libraries, the search query and inclusion/exclusion criteria.

2.1. Research questions

The purpose of this systematic literature review is to provide an overview of the research reported in the field of SECO. In this overview, we intent to address the following research questions:

RQ 1: *How is the term 'software ecosystem' defined?*

In order to be able to analyze the field of SECOs, we should first define the SECO as object of study. Thus, the first objective of this work is to provide an overview of how the research community defines the term 'software ecosystem'. We achieve that by looking into the SECO definitions in the literature and comparing them. This will create an understanding of what the research community means by the term SECO.

RQ 2: *What is the research output per year in the SECO field?*

By grouping the literature per publication year we are able to identify possible trends in the research invested in the field of SECOs. An increase in the number of publications per year, for example, would imply the increase in importance of the field while a decrease in the number of publications might have as a possible reason the research in the field reaching a dead end. Analyzing the trends might give an idea of how the importance of the field of SECOs is changing with time.

RQ 3: *What is the type of result that software ecosystem research reports?*

After having defined the term SECO, a question that we want to address is what kind of research this field reports. Therefore, it is of interest to classify the papers according to the contribution they make. From a software engineering perspective, Shaw's classification of research results (Shaw, 2003) has been chosen. The classification contains the following categories:

Procedure or technique: This category includes papers that are providing a concrete and implementable way to solve a SECO problem. The solutions should be in the form of a procedure or technique that can be applied and not general rules of thumb or reported experiences. For example, Kazman et al. (2012) analyze a series of traditional software design and software architecture principles and methods in the perspective of the SECOs (or software-intensive ecosystems as they are called in the paper). This results in some new or adapted methods for the software design and architecture of these software-intensive ecosystems.

Qualitative or descriptive model: Papers using models based on qualitative analysis of data or well argumentation of existing cases. Papers in this category provide an analytical or descriptive model for the problem area. As an example the analysis of two different kinds of SECO: the "as-a-service" and "on-premise" software ecosystems that derived from a comparative study of two existing SECOs presented in Hilkert et al. (2010).

Empirical model: This category includes papers that use models derived from the quantitative data collection of the problem area. A paper of this category studies empirical data and concludes some analysis or predicting model. For example, Yu et al. (2008) extract information from open source systems to assess the evolvability of software.

Analytic model: Papers using models based on automatic or mathematical manipulation for solving a specific problem. For example the paper of Capuruço and Capretz (2010) that propose a prediction of recommendations and interaction between the members of a social ecosystem based on a mathematical analysis of the member relationships.

Tool or notation: A tool or notation created or implemented applying some method or technique. For example, a tool for recovering components and their relationships in free or open source projects, proposed by Lungu (2008)

Specific solution, prototype, answer, or judgment: Papers documenting a complete solution, evaluation of a theory or comparison of different theories based on a software engineering problem. The result is addressing a specific problem. An example would be Pettersson and Gil (2010) who address reusability and adaptability issues in mobile learning systems

Report: Papers documenting knowledge and experience obtained, rules of thumb or checklists but not systematic enough to be a descriptive model. For example, the analysis of the hybrid business and revenue models that software companies can have (Popp, 2011).

RQ 4: *What is the role of architecture in software ecosystem research?*

For single systems, software architecture is seen as important in determining the quality of a system being built (Bass et al., 2003; Hansen et al., 2011). In relation to this, we analyze the extent to which SECO literature stresses software architecture. We evaluate the literature in whether it is documenting any considerations towards SECO software architecture. In doing so, our concept of software architecture is in line with Bass et al. (2003):

> "The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them."

We here extend the definition to concern software ecosystems, i.e., we define 'software ecosystem architecture' as the structure or structures of the software ecosystem in terms of elements, the properties of these elements, and the relationships among these elements. The SECO elements can be systems, system components, and actors. Relationships then include software architecture-related relationships as well as actor-related relationships such the relationship between two actors.

RQ 5: *How is the connection between research and industry in the area of software ecosystems?*

It is of interest to know how close industry and research are in the field of software ecosystems. Research benefits from realism of problems when connected to the industry while industry arguably may become more innovative and efficient when connected to research. In the case of SECOs research results are more valid when they are concerning existing SECOs, while studies of problems in existing SECO can help the industry improve.

We investigate how connected the research world is with the industry by examining how much of the literature has focused on real-world SECOs. We accept that a paper has focus on a real-world SECO when it either presents an existing SECO as an object of study or uses the data from the study of one to support a claim or result. For example, this could be a paper that is deducting information of the external actors of an ecosystem by studying the relationships between the actors of one or more existing SECOs. However, we do not include papers that merely mention a SECO, e.g., in order to support their definition of SECOs, and that thus present no study of the SECO.

2.2. Defining the literature body

The strategy for collecting the relevant literature is twofold: (i) a keyword search in a list of scientific libraries and (ii) the collection of the papers from the SECO workshop series.

With respect to (i), the scientific libraries included in the search are:

1. The ACM Digital Library[4]
2. IEEE Explore[5]
3. Springer Verlags' digital library, SpingerLink[6]
4. ScienceDirect.[7] An online collection of published scientific research operated by the publisher Elsevier.
5. Thomson Reuters' Web of Science.[8] An online academic citation index.

The literature extraction consists of two separate keyword searches with the search terms "software ecosystem" and "software ecosystems" in the libraries above. The search query is intentionally kept simple so we can extract the maximum number of papers containing the terms. We specifically define SECO(s) as the keyword to underline the differentiation of the field of software ecosystem from different ecosystems like business, digital or social. The borders of the SECO field can be sometimes vaguely defined especially when overlapping with other kinds of ecosystems. For example some SECOs in the literature can also fit in a digital ecosystem definition and there are several studies on business ecosystems that produce software. The purpose of this work, however, is to study software ecosystems and any possible intersections with other ecosystems should be studied from the SECO point of view. Therefore, this study does not include studies on other kinds of ecosystems.

With respect to (ii), we include the papers from the International Workshops on Software Ecosystems (IWSECO).

The selected literature body collected from both (i) and (ii) should commit to a set of inclusion criteria:

- The literature should address software ecosystems as an area of research, either main or secondary. Therefore, the keywords "software ecosystem(s)" should exist as a whole and continuously in at least one of the fields: title, keywords or abstract. Additionally, possible composites of the keywords should be examined, e.g., software-intensive ecosystems.
- Be research papers, i.e., being published in a scientific peer reviewed venue.
- Be written in English.
- Have a document body that is more than one page long.

Consequently, the literature does not contain books, extended abstracts, presentations, presentation notes, keynotes or papers written in other language than english.

The literature body is the results of the following steps:

1. Collecting all the literature. The literature collection is the combination of the scientific library search and the IWSECO papers. The library search, at this point includes a search of the keywords in the whole text body in order to include the maximum amount of papers.
2. Applying inclusion/exclusion criteria. The literature collection resulting from the previous step are searched for the keywords in the fields title, abstract, keywords.
3. Verifying rejected papers. The rejected literature from the previous step is searched for only the terms "ecosystem(s)" and "software" in the fields title, abstract, keywords and evaluated if they are related literature. This would avoid rejecting papers with different combinations of the keywords, for example "software-intensive ecosystems".
4. Verifying included papers. The included literature that resulted from the two previous steps is verified manually by reading the abstract and conclusion. In this step, we make sure that the papers included in the review provide results that are directly or indirectly related to the field of SECO.

3. Collecting the literature body

To obtain the literature body of our review, we apply the systematic literature review (SLR) protocol described in Section 2 with the extraction date of June 11, 2012. The four steps for defining the literature body described in Section 2.2 can be seen in Table 1. The literature collection starts with 420 papers extracted from the libraries. All the IWSECO papers are included in this collection. After applying the inclusion/exclusion criteria, we reject 297 paper. Out of the 297 rejected, we apply step 3 and included six papers with key words "open ecosystems", "software-intensive ecosystems", "ERP ecosystems", "information ecosystem", "source code ecosystems", "Eclipse ecosystem". In step 4 we went through 129 papers (123 from step 2 plus 6 from step 3) and find 90 papers relevant. We contribute the high number of rejected papers in step 2 to two reasons: (i) some libraries would search in the whole paper text body and thus retrieve papers mentioning SECO but not reporting research on that field and (ii) Science Direct does not recognize the quotation marks in "software ecosystem" or "software ecosystems" so it retrieves results that the words are not adjacent to each other but in different locations in the texts, therefore there were many papers not related to software engineering. We also note that from the six papers selected in step 3, only one (Kazman et al., 2012) is part of the included papers.

During the data extraction process, we read the papers found relevant and extracted interesting information and information needed to address the research questions. The information extraction is in the form of descriptive text enclosed by identifying labels for automated sorting. In continuation, a set of custom scripts export the requested information.

Table 1
The steps and included papers to define the literature body.

Step	Nr of papers
1. Collecting the literature	420
2. Applying inclusion/exclusion criteria	123
3. Verifying rejected papers (included)	6
4. Verifying included papers	90

[4] http://dl.acm.org/.
[5] http://ieeexplore.ieee.org.
[6] http://www.springerlink.com/.
[7] http://www.sciencedirect.com/.
[8] http://apps.webofknowledge.com/.

4. Analysis

In this section we analyze the literature and the results of the review. The section is organized according to the research questions in Section 2.1.

4.1. Defining SECO

During this literature review, we obtained an overview of the general field referred to as *software ecosystems*. One of our initial aims was to define the term SECO by summarizing the definitions in the literature. Looking into the literature, our first remark is that we found a large number of papers (40 out of the total of 90) that did not define the term SECO. This is, either because the authors are basing their work on previous research (own or not) that would provide the background and definition or because the main focus of the paper is not in the general field of SECO. For example, Bosch (2010a) is not providing any definition, but he is referring back to his own work (Bosch, 2009) where he provides a definition and more detailed analysis of the field. On the other hand, Popp (2011) defines the business and revenue models for SECOs. In his paper, he is providing definitions for the business and revenue models that is the main focus, instead of a definition of a SECO. This, however, does not make it of less value to the research field of SECOs.

Taking the papers that provide a definition, we notice that few of them are defining the SECO with their own words. Two of these papers are also citing more definitions from the literature along with their own. The rest of the papers, are defining the field by using one or more definitions from the existing literature. When we analyzed the definitions, we found that we can group the quoted definitions in four groups according to the source of the definition:

Messerschmitt and Szyperski (2005) is the oldest definition of SECO in the found literature referring to the book on SECO published in 2005.

> "Traditionally, a software ecosystem refers to a collection of software products that have some given degree of symbiotic relationships." (Messerschmitt and Szyperski, 2005)

Jansen et al. (2009b) mainly refer to the following definition:

> "We define a software ecosystem as a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently under-pinned by a common technological platform or market and operate through the exchange of information, resources and artifacts." (Jansen et al., 2009b)

Bosch (2009) and Bosch and Bosch-Sijtsema (2010b,c) provide two definitions in their papers. The papers quoting his definitions are taking one of the following:

> " A software ecosystem consists of the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions." (Bosch, 2009)
>
> "A software ecosystem consists of a software platform, a set of internal and external developers and a community of domain experts in service to a community of users that compose relevant solution elements to satisfy their needs." (Bosch and Bosch-Sijtsema, 2010b,c)

Lungu et al. (2010a) are presenting a different definition of the SECOs that is adopted by a number of papers:

> "A software ecosystem is a collection of software projects which are developed and evolve together in the same environment." (Lungu et al., 2010a)

In Table 2 we show the different groupings and the papers belonging to each group. The in the column Papers refer to the literature body listed in Appendix A.

Not surprisingly, if we look at the definitions we can see that they have two things in common: they concern software in some form (software systems, products, services, or a software platform) and they are all including some kind of relationships either "symbiotic", "common evolution", "business" or "technical". If we look at what perspective the authors have in the definitions, we note that Messerschmitt and Lungu et al. have a pure technical perspective by talking about software and its symbiosis/co-existence, while Bosch et al. and Jansen et al. include, apart from the technical, a social and business perspective to their definition and the symbiosis is not only on the technical level. Taking the two wider-perspective definitions of Bosch et al. and Jansen, which are referenced by the majority of the papers that provide a definition for SECO (65%), we can identify three main elements in their definitions:

Common Software The software appears either as a "common technological platform" (Jansen et al., 2009b), "software solutions" (Bosch, 2009) or "software platform" (Bosch and Bosch-Sijtsema, 2010b,c)

Business This is expressed as either "a set of business" (Jansen et al., 2009b), "business ecosystem" (Bosch, 2009), a "community of users that have needs to be satisfied" (Bosch and Bosch-Sijtsema, 2010b,c). In this element, the term "Business" is implying a wider sense than the profit or revenue models. This element also includes possible benefits other than financial revenues, e.g., the benefits an actor would get from the involvement in an free or open source project.

Connecting Relationships "a set of businesses (...) together with the relationships among them " (Jansen et al., 2009b), "actors in the associated social ecosystem" (Bosch, 2009), "community of domain experts" and "community of users" (Bosch and Bosch-Sijtsema, 2010b,c)

Combining the definitions above with the three elements identified, we define a software ecosystem as the interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services. Each actor is motivated

Table 2
The papers belonging to each group of SECO definition.

Definition	Papers	Total
Not available	[19, 1, 45, 39, 43, 2, 5, 6, 9, 11, 48, 49, 59, 52, 51, 54, 42, 53, 36, 46, 31, 22, 21, 35, 33, 26, 55, 32, 24, 63, 82, 74, 75, 69, 62, 64, 70, 78, 67, 83]	40
Jansen et al.	[3, 4, 10, 16, 13, 28, 37, 44, 14, 29, 12, 6, 27, 87, 86, 72, 76, 71, 61, 65, 66, 60, 90, 84]	24
Bosch et al.	[40, 41, 10, 13, 20, 23, 44, 14, 17, 12, 89, 77, 79]	13
Own	[38, 8, 30, 58, 56, 47, 12, 34, 73]	9
Lungu et al.	[7, 15, 18, 80, 68, 81]	6
Messerschmitt et al.	[40, 50, 37, 57, 85]	5

Table 3
Papers published per year.

Year	Papers	Total
2007	[31, 46, 57]	3
2008	[50, 53, 54]	3
2009	[9, 10, 11, 20, 6, 42, 51, 52, 56, 58]	10
2010	[1, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23,24, 27, 28, 29, 30, 33,34, 35, 36, 37, 38, 39,41, 43, 45, 47, 48, 49,55, 59]	32
2011	[2, 3, 4, 5, 6, 7, 8, 26,32, 40, 44, 87, 89, 88,73, 72, 74, 68, 71, 69,64, 61, 65, 70, 79, 66,60, 67, 83, 85, 90, 84]	32
2012	[63, 80, 86, 82, 76, 77,75, 62, 78, 81]	10

by a set of interests or business models and connected to the rest of the actors and the ecosystem as a whole with symbiotic relationships, while, the technological platform is structured in a way that allows the involvement and contribution of the different actors. In other words, the SECO provides possibilities for the actors to benefit from their participation in the ecosystem. The types of benefits might vary depending on the actor and the nature of the ecosystem. In a commercial ecosystem the actors might gain direct revenues, e.g., developers making apps for iPhone and selling them in the App Store, while in a non-commercial ecosystem the actors might participate for non-monetary benefits (fame, knowledge, ideology and so on), e.g., the developers contributing to Apache. Additionally, the actors' relationships to the ecosystem as a whole are of mutual interest (mutualism): the actors' benefits increase by the thriving of the ecosystem and the ecosystem benefits by increased actor activity. The relationships among the actos in a SECO, on the other hand, are characterized by the wider spectrum of symbiotic relationships. Depending on the actors and their activity, two actors might have mutual benefits (mutualism), be in direct competition (competition/antagonism), be unaffected (neutralism) or one being unaffected while the other is benefiting (amensalism) or harmed (parasitism) by their relationship.

When looking at the rest of the papers, we note that there is a number of papers that assist in the conceptualization of the field in a wider sense than just providing the definition of SECOs. These papers are used as a conceptual base of succeeding work. In this concept, Bosch (2009) proposes a taxonomy where he divides SECOs in three categories: operating system-centric, application-centric and end-user programming software ecosystems. In continuation he discusses the steps needed for the transition to a SECO and implications this transition might have. Jansen et al. (2009a), apart from providing the definition for SECO seen above, propose three scopes to study SECOs that are also explained briefly in Boucharas et al. (2009): an external view on ecosystems that studies the SECOs themselves and the markets around them, an internal view of a SECO that is focusing on software supply networks and their relationships, and an organization-centric perspective that studies the actors and their relationships. Campbell and Ahmed (2010) propose a view of SECO consisted of three dimensions: business, architectural and social. Dhungana et al. (2010) make a comparison of the SECO with biological ecosystems from the perspective of resource management and biodiversity and underline the importance of diversity, monitoring of health and supporting social interaction for the field of SECO.dos Santos and Werner (2011b) collect the concepts appearing in the papers from IWSECO 2009 to 2010 and organize them in three views: SECO architecture, SECO strategies and tactics and SECO social networks. Finally, Barbosa and Alves (2011) conduct a systematic mapping in the field of SECO and categorize the research in eight fields unfolded around open source software, ecosystem modeling, and business issues.

4.2. Yearly activity

Another point of study in this work, is the analysis of the year of publication. We order the papers according to their publication year as can be seen in Table 3. The literature on SECO starts in 2007 (although (Messerschmitt and Szyperski, 2005) dates back to 2005, it was excluded from this study for being a book and not a research paper). The first two years – 2007 and 2008 – provide an equally low number of papers. However, an increase appears in 2009 and continues to 2010 with 2010 and 2011 having the same amount of papers.

The increase of papers gives us a clear sign that the field of SECO is gaining in importance among the published research. This is also underlined with the establishment of a workshop dedicated to SECOs, the International Workshop on Software Ecosystems (IWSECO), in 2009. While this does not give insight into software ecosystems in themselves, it stresses the potential significance of the concept.

4.3. Research results

As noted in research question in Section 2.1, it is of interest to examine what kind of results the papers are reporting. We have classified the papers in the categories listed in research question in Section 2.1 and can be seen in Table 4. As it can be seen from the table, the majority of the papers fall under the Report category. This means that these papers have as contribution knowledge and experience obtained, rules of thumb or checklists or interesting observations but they are not systematic enough, nor generic enough to be applied to different domains or too abstract to provide a concrete contribution. An example of a paper falling under this category is the paper by Dhungana et al. (2010) that compares

Table 4
The papers grouped according to the result groups.

Result	Papers	% of total
Report	[19, 43, 2, 6, 10, 8, 59, 52, 56, 51, 36, 46, 31,20, 35, 12, 26, 32, 24,14, 29, 33, 17, 87, 73,86, 82, 76, 77, 71, 75,64, 61, 65, 70, 78, 66,67, 83, 85, 84]	44
Tool or notation	[9, 48, 58, 54, 15, 22,47, 6, 80, 68, 69, 62, 79,81]	15
Procedure or technique	[40, 41, 5, 11, 30, 53, 16, 13, 28, 18, 14,17]	13
Qualitative or descriptive model	[38, 39, 3, 4, 7, 21, 37, 29, 34, 74]	11
Empirical model	[45, 50, 44, 57, 55, 27, 89]	8
Analytic model	[1, 42, 63, 72, 90]	5
Specific solution	[49, 23, 33, 88]	4

Table 5
The papers according to the SECO Architecture groups.

SECO architecture group	Papers	% of total
SECO SE	[40, 19, 45, 38, 39, 43, 11, 49, 58, 36, 46, 15, 22, 47, 35, 29, 17, 57, 87, 63, 68, 69, 70, 79, 66, 81]	35
SECO business and management	[2, 4, 5, 10, 58, 56, 16, 20, 23, 37, 44, 14, 33, 12, 6, 27, 87, 88, 86, 76, 77, 71, 75, 62, 66, 60, 67, 83, 90]	39
SECO relationships	[3, 4, 6, 30, 13, 31, 21, 33, 17, 87, 73, 86, 82, 72, 76, 61, 65, 85, 84]	26

SECOs to the natural ecosystem and reports observations and a research agenda. This paper does not report any concrete method of some kind and the data used is not systematic enough for the paper to be included in the qualitative model.

Looking at the distribution, we note that the category with the most papers after Report is Tool or Notation. The papers of this category are implementing tools or notations that are mostly using data from FOSS SECOs. This, as we will discuss more in Section 4.5, is related to the fact that FOSS SECOs provide access to a lot of technical data, e.g., commit history or bug reports that are not easy to access in proprietary SECOs. The third category, Procedure or Technique, includes papers that report an implementable technique to solve a specific task. For example the paper by Fricker (2009), that proposes a technique for requirement management in SECOs.

When examining the percentage of papers that fall under each category, we can make the following observations. The field of SECOs is a new research field, with the first papers appearing in 2007. This implies that there is an amount of research resources spent in defining the field and its limits, for example the papers analyzed in Section 4.1. In addition, as it is shown in Section 4.5, there is a relatively small amount of research spent in examining SECOs in the industry. These two reasons result in the Report category having a bigger percentage to all the other categories. Additionally, we recognize that the field of SECO is wide and can have multiple research perspectives, such as software engineering (SE), social networks or technical management. In connections to this, there have been several papers focusing on some specific aspect of the field providing specific and implementable techniques. This potentially explains the high percentage in the Tool or Notation and Procedure or Technique categories.

4.4. SECO architecture

To address RQ in Section 2.1, we separated and analyzed the papers that are addressing the SECO architecture as defined in the research question. During the analysis of the papers, we could identify three logical groups of SECO architecture papers. Table 5 shows the distribution of the papers according to their main research focus. Below we elaborate on the three SECO architectural groups describing them in more detail. Papers used in the description of a group might not represent their main research focus.

4.4.1. SECO software engineering

Software ecosystems, having as a product one or several software systems have problems that belongs to the software engineering field. A part of the SECO literature is focusing on SE either by using SE practices directly or by adapting existing SE practices to the SECO context. This category consists of papers focusing on more technical issues related directly or indirectly to the technological platform of a SECO. It contains 26 papers, i.e., 35% of the literature focusing on SECO architecture aspects.

One important aspect of this category is software architecture. The software architecture of a SECO should support the nature of the ecosystem (i.e., be adapted to the needs of the specific SECO), follow the SECO management, business rules and restrictions and allow the integration and existence of multiple functionality in a secure and reliable manner. A modular and flexible architecture would allow integration and interoperability of the developed software (Viljainen and Kauppinen, 2011; Bosch, 2009). Interfaces allow external development on a SECO platform. The stability and translucency of the platform interfaces are essential for the component integration and interaction (Cataldo and Herbsleb, 2010; Bosch, 2010a). Changes to existing interfaces or components might create inconsistencies to dependent components (Robbes and Lungu, 2011; Lungu et al., 2010a,b). Process-centric approaches are not effective in managing large scale software, instead system architecture should be used as a coordination mechanism (Bosch and Bosch-Sijtsema, 2010a). Constantly evolving software requires the adaptation of the software development processes. Development should be integration-centric, independent deployment and releases should be organized in a release grouping and release train fashion (Bosch and Bosch-Sijtsema, 2010b; Bosch, 2010a). Architectural design and analysis techniques are based on a set of principles as identifying business goals, describing architectural significan requirement, tactics and architectural evaluation. These principles are used in defining the software architecture of a SECO (Kazman et al., 2012).

Apart from software architecture, in the wider SE related subjects, requirement elicitation appears as an interesting challenge in the SECO concept as the stakeholders are multiple and distant from the central ecosystem management. The use of "requirement value chain" is proposed to propagate requirements (Fricker, 2009, 2010).

4.4.2. SECO business and management

This category contains papers focusing on the business, organizational and management aspects of SECOs. Independently of how each SECO is organized, there is an organizational and management entity that is responsible for monitoring, operational and decision making part of the SECO whether it being a proprietary company, an open source community or a hybrid of the two. This category is sub-divided into two groups: organizational & management and business.

The organizational and management group includes papers that are focusing on the organizational actions in a SECO. These actions are initiated from decisions, rules and processes or controlling mechanisms. The main activities of this group are summarized in: monitoring the SECO, evaluating and decision making, and taking actions.

In order to ensure that a SECO is functioning well, specific measurements need to be introduced that would provide an overview of the state of the SECO while at the same time raise attention for actions and allow comparison of SECOs. The literature is referring to the concept of the *health of a software ecosystem* (van Ingen et al., 2011; van Angeren et al., 2011; van den Berk et al., 2010; dos Santos and Werner, 2011a,b; Kilamo et al., 2012; Jansen et al., 2012, 2009a; Viljainen and Kauppinen, 2011; Mizushima and Ikawa, 2011; McGregor, 2010; Dhungana et al., 2010; Boucharas et al., 2009). This concept has been introduced by Iansiti et al. as a way to measure the performance of a business ecosystem (BECO). In more detail they measure the "*extent to which an ecosystem as a whole is durably growing opportunities for its members and those*

who depend on it" (Iansiti and Levien, 2004a) and inspired from biological ecosystems define the health of a (business) ecosystem as an analogy to robustness, productivity and niche creation (Iansiti and Levien, 2004b,a). These studies, although excluded from the collected literature, are referenced by the majority of the literature elaborating on SECO health (van Ingen et al., 2011; van Angeren et al., 2011; van den Berk et al., 2010; Kilamo et al., 2012; Jansen et al., 2012, 2009a; Viljainen and Kauppinen, 2011; Mizushima and Ikawa, 2011; McGregor, 2010; dos Santos and Werner, 2011b; Boucharas et al., 2009). An additional study on the health of business ecosystems that is referenced by several papers of the literature (van Angeren et al., 2011; van den Berk et al., 2010; Kilamo et al., 2012; Jansen et al., 2012, 2009a) that are elaborating on SECO health, is the paper of den Hartigh et al. (2006) that, based on the Iasiti et al studies mentioned above, applies health measurement to Dutch IT business ecosystems. In the SECO field, van den Berk et al. (2010) base their work on BECO health to create a strategy assessment model.

The proper evaluation of SECO measurements, such as health, supports and encourages correcting or improving actions in the SECO. This requires a management entity that would have the power and possibility to apply changes both in the technical but also in the organizational aspects of the SECO. To our knowledge there is no study in the SECO literature on the different management entities and the decision making mechanisms applied to drive the SECO. This might be because of high variability in management models or the disclosure of information in proprietary SECOs. It would be possible to study the decision making mechanisms of a FOSS project where changes are applied, e.g., based on online member voting, but it is challenging to study how a proprietary SECO canalizes information from the peripheral actors, evaluates this information and decides on actions based on that.

After monitoring the SECO and concluding in a set of decisions, a next step is to execute these decisions. One of the ways of applying actions that appears in the literature is communication. A clear view on the direction that the ecosystem would evolve and the communication of this view to the ecosystem actors and involved parties is underlined as a necessity (Bosch, 2009; Viljainen and Kauppinen, 2011). Creating *roadmaps*, *visions* or long-term *strategic planning* of the ecosystem allows the actors to plan, in their turn, their activity in the ecosystem and align their business models with the SECO roadmaps (Kakola, 2010; Bosch, 2009; Hanssen, 2011; Viljainen and Kauppinen, 2011; Jansen et al., 2012; van den Berk et al., 2010). At the same time the ecosystem can set the requirement of the SECO actors to commit to the published roadmaps (Bosch and Bosch-Sijtsema, 2010b,c). From a more practical perspective, the ecosystem orchestrators can organize the component composition by providing a long-term plan of organized releases in a *release management* or *release trains* that the actors can coordinate with (Bosch and Bosch-Sijtsema, 2010b,c; Fricker, 2010; Jansen et al., 2012; van den Berk et al., 2010; van der Schuur et al., 2011; Bosch, 2009). Bosch and Bosch-Sijtsema (2010c) analyzed the concept of *release grouping* where different groups of components are released in different times allowing less coordination and communication overhead. Kilamo et al. (2012) introduce the *release readiness assessment* where proprietary software is assessed on its ability to be released as open source/ open ecosystem.

An important part of the SECO business and management category is related to the business perspective of the ecosystem. As explained in the definition analysis, the business perspective is important as without a solid business and business model serving the SECO and its actors, the SECO might lose its actors to competitive businesses or ecosystems and risk extinction. It is essential to underline that the business and business model as mentioned here do not necessarily imply monetary benefits. The business model that would serve the SECO actors, as mentioned in the definition, might imply value in other forms, for example fame or experience in the case of a FOSS SECO actor. The same applies to the SECO itself. A SECO might include other benefits than revenues in its business model. An example would be advantage over competitors or "visibility within the market" (van Angeren et al., 2011). This implies that the traditional software company business models where the revenues are a result of software license selling cannot be fully applied in the ecosystem concept. Popp (2011) provides an analysis of business models that are applied in three ecosystems and makes a separation between the business models and the revenue models of a SECO. He underlines the importance of revenue models and states that "*revenue models (. . .) often containing one or more non-monetary compensations, can be a source of competitive advantage*" (Popp, 2011). Burkard et al. (2012) refer to revenue models from two perspectives: actors or niche players provide their products for a fee and the SECO orchestrator or hub requires a fee from the actors. This fee can be base either on fixed or variable price models.

Although, selling software licenses might not be a main revenue venue for a SECO, the issue of software licenses is still of interest in the SECOs. SECOs collect code developed by different developers or companies with different policies and many times even in an combination of proprietary and open source. Addressing or avoiding possible intellectual property right (IPR) or licensing violations would ease the software integration, allow possible reuse that might lead to more niche creation, clarify possible business models and avoid legal complications that demand heavy resources. Licensing and IPR issues appear in a number of papers (Alspaugh et al., 2009; Jansen et al., 2012, 2009b; Mizushima and Ikawa, 2011; te Molder et al., 2011; Kilamo et al., 2012; Scacchi and Alspaugh, 2012) in the literature. In relation to this, Alspaugh et al. (2009) and Scacchi and Alspaugh (2012) discuss the issue of software licensing in open architecture systems, recognize changes in licenses on different versions of the same component or in the evolution of a software system and propose a structure for modeling software licenses. Mizushima and Ikawa (2011) analyze the IP management process of Eclipse called the "Eclipse Legal Process" and state that this process was a reason for vendors to join Eclipse. Anvaari and Jansen (2010) analyze the mobile software platforms and evaluate their level of openness taking into consideration also their licensing policies. Finally, Popp (2011) names three roles in the intellectual property (IP) business utilization: the IP distributors that sell IPR from the inventors or usage rights to the customers, the IP lessors that "rents" IPs or products of IP (e.g., software) for a specific time and the IP brokers that matches the needs of an IP requestor to an IP owner. For example an IP broker might facilitate a startup software company to find software vendors.

4.4.3. SECO relationships

An open technological platform in combination with a set of management processes and business models, cannot create a SECO without the social aspect. A community, social network or a set of actors weaved around a platform and sets of rules communicating and interacting both among themselves and with the platform is essential. Because of the existence of this interaction, the software architecture of the platform has to be designed with different considerations than a proprietary platform. The management process, business models and IPR issues become more complicated while at the same time the evolution of the system is faster and towards several directions while the SECO gains privileged position in the market. There are several actors that might be part of a SECO. The following list gives an overview of the most common actors encountered in the literature.

Orchestrator [9], "keystone (player, organization)", [10] "hub",[11] "shaper",[12] "management (unit)",[13] or "platform owner"[14] is a company, department of a company, actor or set of actors, community or independent entity that is responsible for the well-functioning of the SECO. This unit is typically managing the SECO by running the platform, creating and applying rules, processes, business procedures, setting and monitoring quality standards and/or orchestrating the SECO actor relationships.

Niche player [15]"influencer",[16] or "component developer/builder/team",[17] is the SECO actor that contributes to the SECO by typically developing or adding components to the platform, producing functionality that customers require. This actor is part of the SECO and complements the work of the keystone by providing value to the ecosystem. Depending on the management model of the ecosystem the niche players might influence the decision making in the management of the SECO.

External actor [18]"external developer (team)",[19] "third party developers/community",[20] "external parties",[21] "external partner",[22] "external entities",[23] "participant",[24] or "external adopter",[25] is the actor (company, person, entity) that makes use of the possibilities the ecosystem provides and thus providing indirect value to the ecosystem. This actor is external to the SECO management and usually has an activity limited to the actor's interest. Depending on the nature of the ecosystem, the external actor might be developing on top of or parallel to the SECO platform, identify bugs, promote the SECO and its products or propose improvements. This type of actor includes the role of the participant or follower in FOSS SECOs. An actor that is member of the SECO with either participation of limited responsibility or simply observing the evolution of the SECO from the inside.

Vendor "independent software vendor (ISV)",[26], "reseller" or "value-added reseller (VAR)",[27] is mainly the company or business unit that makes profit from selling the products of the SECO to customers, end-users or other vendors/VARs. The products might be complete integrations, components, selling or leasing of licenses or support agreements. A vendor that is modifying the SECO product by, e.g., adding functionality or combining different components together is called VAR.

Customer or "end user" is the person, company, entity that either purchases or obtains a complete or partial product of the SECO or a niche player either directly from the SECO/niche player or through a vendor/VAR.

A different characterization of the social network of a SECO appears in (Jansen et al., 2012; Scacchi and Alspaugh, 2012) where they characterize the SECO niche as a software supply network of producers, integrators and customers.

An interesting perspective of SECO relationships is the actor participation model that SECOs follow. Different ecosystems apply different models for allowing actors to contribute to the ecosystem. These models are many times related to the nature of the platform and to what extent it allows/supports different kinds of collaboration, but mostly to the business model behind the ecosystem. To explain this better, we take the actor participation model of three ecosystems as an example: a traditional FOSS project that is often open to any participant willing to join, the Eclipse ecosystem where developers can join freely but have to go through the Eclipse Legal Process every time they commit code (Mizushima and Ikawa, 2011) and the the Open Design Alliance (ODA) where actors have to pay an annual fee to be part of the ecosystem (van Angeren et al., 2011). The *openness* or *closeness* of a SECO describes how easy it is for an actor to participate in an ecosystem. The measurement of the openness of a SECO is an interesting perspective that affects the social network of an ecosystem. As already mentioned, the level of openness depends on parameters outside of the SECO social network perspective, however, it is analyzed as part of this perspective since it affects heavily the social networks. te Molder et al. (2011) claim that the openness and closeness of a platform is not binary, but there are many different levels. In their paper they introduce the concept of "clopeness" and propose a model for assessing the clopeness of a SECO. Jansen et al. (2012) state that the complicity of opening or closing the SECO as "multi-facet and cannot be judged without extensive study". They also explain that the benefits of opening up the ecosystem are often not clear, while a post-evaluation of whether the ecosystem was ready for the changes will be reflected in the SECO health after the changes have been applied. Finally they make a separation between the supply and demand of a SECO and mention that a SECO can choose to open either of them or both.

In the software supply network, Riis and Schubert (2012) analyze how the relationships evolve in an ERP SECO when the SECO vendor (orchestrator) is pushing an upgrade to a newer version. It is notable that the relations can be push-oriented, i.e., the orchestrator pushes a new version to the ISVs and VARs and eventually the customer, but also pull-oriented, i.e., the customer requests an older version from the ISVs/VARs end eventually the orchestrator. Jansen et al. (2012) referring to Popp (2010) numbers three distribution channels: (i) direct through VAR, (ii) indirect through

[9] Used in: van Angeren et al. (2011, 2011), Jansen et al. (2009b,a), Hilkert et al. (2010), Idu et al. (2011), van der Schuur et al. (2011).

[10] Used in: van Angeren et al. (2011), Burkard et al. (2012), Campbell and Ahmed (2010), Hanssen (2011), Jansen et al. (2009a, 2012), Kabbedijk and Jansen (2011), McGregor (2010), Pettersson et al. (2010), Riis and Schubert (2012), Viljainen and Kauppinen (2011), Idu et al. (2011), dos Santos and Werner (2011a,b), te Molder et al. (2011), van den Berk et al. (2010), van Ingen et al. (2011), van der Schuur et al. (2011).

[11] Used in: dos Santos and Werner (2011a,b), Burkard et al. (2012), Hilkert et al. (2010), Riis and Schubert (2012), van den Berk et al. (2010).

[12] Used in: Jansen et al. (2009a), Viljainen and Kauppinen (2011), van der Schuur et al. (2011)

[13] Used in: Campbell and Ahmed (2010).

[14] Used in: van Angeren et al. (2011).

[15] Used in: Jansen et al. (2009a), Viljainen and Kauppinen (2011, 2011), dos Santos and Werner (2011a,b), Burkard et al. (2012), Yu and Deng (2011), Kabbedijk and Jansen (2011), Riis and Schubert (2012), te Molder et al. (2011), van den Berk et al. (2010), van Ingen et al. (2011), van der Schuur et al. (2011).

[16] Used in: dos Santos and Werner (2011a,b), van den Berk et al. (2010).

[17] Used in: Jansen et al. (2009a, 2012), Viljainen and Kauppinen (2011, 2011), Bosch and Bosch-Sijtsema (2010c), Bosch (2009).

[18] Used in: Pettersson and Gil (2010), Hansen et al. (2011), Pettersson et al. (2010).

[19] Used in: Bosch (2010a, 2009, 2010b), Pettersson et al. (2010), dos Santos and Werner (2011a,b), Bosch and Bosch-Sijtsema (2010b,c,a), Jansen et al. (2012), Draxler and Stevens (2011), Kilamo et al. (2012), Viljainen and Kauppinen (2011), Scacchi and Alspaugh (2012), van Ingen et al. (2011), Weiss (2011).

[20] Used in: Anvaari and Jansen (2010), Bosch and Bosch-Sijtsema (2010b,c), Bosch (2009, 2010b), Campbell and Ahmed (2010), Dhungana et al. (2010), Hanssen (2011), Jansen et al. (2009a, 2012), Mizushima and Ikawa (2011), Seichter et al. (2010), Viljainen and Kauppinen (2011).

[21] Used in: Bosch and Bosch-Sijtsema (2010b,c).

[22] Used in: Bosch (2010b), Draxler and Stevens (2011).

[23] Used in: Campbell and Ahmed (2010).

[24] Used in: Jansen et al. (2009a).

[25] Used in: Viljainen and Kauppinen (2011).

[26] Used in: Jansen et al. (2009b,a, 2012), Bosch (2009), Boucharas et al. (2009), Draxler and Stevens (2011), Hilkert et al. (2010), Hunink et al. (2010), Riis and Schubert (2012), te Molder et al. (2011), van den Berk et al. (2010), Viljainen and Kauppinen (2011), Scacchi and Alspaugh (2012), Janner et al. (2008).

[27] Used in: Riis and Schubert (2012), Jansen et al. (2012, 2009b,a), Janner et al. (2008), Boucharas et al. (2009), Hanssen (2011), Popp (2011).

Table 6
The papers using existing SECOs.

SECO type	Papers	% of total
Proprietary	[45, 41, 2, 4, 10, 30, 54, 53, 16, 37, 44, 33, 6, 26, 87, 63, 82, 72, 62, 65]	22
FOSS	[6, 7, 48, 51, 50, 42, 31, 18, 57, 27, 89, 88, 73, 80, 76, 74, 68, 77, 75, 64, 70, 66, 60, 67, 83, 81, 85, 84]	31
No SECO	[40, 19, 1, 38, 39, 43, 3, 5, 9, 11, 8, 49, 59, 58, 52, 56, 13, 36, 46, 20, 23, 15, 22, 47, 21, 28, 35, 14, 29, 17, 12, 34, 55, 32, 24, 86, 71, 69, 61, 78, 79, 90]	47

service organization and (iii) direct to customer. Yu et al. (2008), Yu (2011) adopt the natural ecology types of symbiotic relationships to software symbiosis: mutualism, where both systems benefit from their relations, commensalism, where one system benefits from the relations while the other is unaffected, parasitism, where one system benefits and the other is harmed, amensalism, where one system is harmed and the other unaffected, competition, where both systems are harmed and neutralism where both systems are unaffected. Although, the symbiotic relations were described in the software symbiosis context rather than the social network, in our perspective, they could also be used to reflect SECO social network relations.

When looking into the niche player relationships, Kazman and Chen (2010) proposes the Metropolis model for the relationships between the actors in a SECO where it is consisted of the kernel that is responsible for platform and fundamental functionality, the periphery that is consisted of the prosumers building on top of the kernel's platform, and the masses that are the end-users. This can be parallelized to the "onion model" (Jergensen et al., 2011; Kilamo et al., 2012) appearing in FOSS projects, where the member involvement is similar to the layers of an onion: a member starts from the external layers having tasks with low responsibility, e.g., translation, and slowly moves to the inner layers gaining responsibilities. In another study of the developer behavior, Kabbedijk and Jansen (2011) studied the interaction of developers within the Ruby Github SECO and noted three different roles: the "lone wolf" that works mainly alone and produces big part of the system used by the rest of the users, the "networker" that is connected to several other developers and the "one day flies" that have created only one popular component without significant activity afterwards.

Communication among the different roles is also of interest. van der Schuur et al. (2011) study how knowledge is transferred within the different roles of a SECO while Fricker (2010) proposes the propagation of information in terms of requirements from the end-users or customers to the ecosystem with the requirement value chains.

4.5. Connection with industry

From the research questions that are mentioned in the beginning of this article, question 2.1 is investigating the use of real-world SECOs in the research. The purpose is to give a view on how close the connection of the research is to the industry. From the data collection process, we have compiled a list with all the papers that are using an existing SECO in their research as an object of study. Analyzing this list, we end up with the results that can be seen in Table 6. Going through the results, we notice that the slight majority of the papers (53%) is using an existing SECO in their research. The existing ecosystems are appearing in mainly two ways: (i) one or more SECOs are studied and the paper publishes study results, conclusions, interesting remarks as it is the case with Hanssen (2011) that describe the transition of a traditional waterfall-based software company to a SECO and (ii) a theory, framework, taxonomy or tool is developed based on literature, hypothesis or experience and then applied to one or more existing ecosystems to prove it, as it is the case in te Molder et al. (2011) where the Clopennes Assessment model is applied to an anonymized SECO to support the theory. In both of the cases, we argue that the use of existing ecosystems as objects of study increases the 'external' validity of the results.

Table 6 is separating the papers that study existing ecosystems in papers studying proprietary and free or open source software (FOSS) ecosystems. We separate the two kinds of ecosystems as they have significant differences. In a strict proprietary ecosystem, the source code and other artifacts produced are protected, as they are the products that would yield revenues to the ecosystem, while new actors would probably have to be certified in some way so they would be allowed to participate in the ecosystem. In a traditional FOSS ecosystem, the actors do not necessarily participate to obtain direct revenues from their activity in the ecosystem, while it is often much easier for an actor to participate in a FOSS than a proprietary SECO, since FOSS SECOs typically do no require any verification of new actors. Naturally, this simplistic way of separating proprietary and FOSS SECOs is only used to underline the differences of the two kinds of ecosystems. A majority of the SECOs would probably be categorized as a hybrid, combining elements from the two kinds. However, in the literature we note that papers studying FOSS SECOs are mostly concerned with problems of technical or social nature, while the papers studying proprietary SECOs include business and strategic problems. This is only natural, since FOSS projects allow the mining and processing of several details (like source code, commit logs, etc.) but they do not necessarily have a clear business model for the whole SECO or the participating actors (or at least it does not apear so in the literature). This underlines the importance of the research focusing on FOSS SECOs to include business and strategic perspectives. On the other hand, papers in the proprietary SECO group can get information about SECO strategies and positioning in the market, but it is harder to get access to proprietary information like source code, developer commits and so on.

Table 7 lists the existing SECOs used in the literature. The literature is studying 43 SECOs in total, out of which, 30 are studied in only one paper each. We note that out of the 12 SECOs studied in more than one paper (in this count we do not include the "Anonymized/not named" category), only two (GX Software and SAP) do not belong to the FOSS group and Eclipse being the most studied SECO (appearing in seven papers). Additionally, 18 out of the 43 studied SECOs are of proprietary nature. We explained this, by the additional challenge posed in gaining access to information in a proprietary SECOs in contradiction to a FOSS where data are usually accessed by mining a publicly available repository.

5. Discussion

The purpose of this study is to provide an overview of the field of software ecosystems by reviewing and analyzing the published literature. This work has been done based on the review protocol explained in Section 2.

In this work we did not include any evaluation of the quality of the relevant literature. The only consideration relating to the quality of a paper is the number of papers within the literature citing this paper, if any. It could be argued that a possible assessment of the quality of the literature could be undertaken to set focus on the gravity each paper should have in the analysis sections, e.g., 4.4.

Table 7
The SECOs appearing in the literature.

SECO name	Papers
Eclipse, Eclipse Foundation	6, 89, 73, 76, 67, 83, 85
GNOME	7, 51, 80, 74
Open Design Alliance	6, 76, 10, 16
Anonymized/not named	65, 82, 45
Brazilian Public Software (BPS)	88, 64, 33
Linux, Linux Kernel	50, 57, 70
Android	27, 66
GX Software	76, 10
Evince	7, 18
FOSS	42, 31
FreeBSD	50, 57
iPhone/iPad App Store	27, 72
SAP	53, 2
Apache Web Server	70
Artop	67
Brasero	7
CAS Software AG	37
CSoft	44
CubicEyes	6
Debian	60
Google Chrome	75
Google	
Gurux	2
Firefox	77
HIS GmbH	75
HISinOne	63
Mac App Store	26
Microsoft	72
Nokia Siemens Networks	2
Nautilus	87
Pharo	81
Ruby	68
S. Chand Edutech	84
SOOPS BV	62
Squeak	54
Symbian	68
TFN 200	67
UniImprove	41
Unity 30	75
US Department of Defense	30
WattDepot	48
WinMob	27
World of Worcraft	89

Apart from addressing the research questions and providing an overview in the field, we also identified several areas that are not covered in the literature body.

As already noted, the field of software ecosystems is not the only field inspired by the natural ecosystems. There has been significant amount of work done in other ecosystems like the business, social or natural ecosystems themselves. The SECO literature does not appear to examine work done in other ecosystems apart from a number of papers mentioned in Section 4. Possible intersections or parallelizations of the fields would allow the use of theories from the other fields or different perspectives in SECO problems.

An important ingredient of the success of an ecosystem is diversity. The differentiation of actors would allow niche creation. Statements similar to this have appeared several times in the literature. However, no concrete studies have been provided to prove a statement of this kind. Technical, organizational, business and social variability in harmonic symbiosis settings could bring more stability and possibly contribute to a healthier ecosystem.

The concept of health of an ecosystem, as explained in Section 4, section has been introduced to SECO from the business ecosystem theory. Measuring the health of an ecosystem would provide large benefits for the SECO industry and research. The health would provide indications on the future of the ecosystem and give possible feedback on applied changes in the ecosystem. However, apart from referring to SECO health, very few studies elaborate, analyze or measure the health of a software ecosystem.

The intellectual property rights and licensing issues are a focus point of a small part of the literature. Finding effective ways to address issues of this kind is of more importance than the attention it has been receiving in the literature. Issues of this kind are of importance both to the organizational perspectives of a SECO – how to organize the development in the ecosystem– but also in the business – how to develop the proper business/revenue models.

Quality assurance (QA) is a field that has also not been efficiently addressed in the literature. The adoption of traditional QA methods might not necessarily work in a SECO, because of the separation of platform and actors. Possibly, the proper QA strategies depend on the orchestration of the ecosystem and solutions might be specific to each SECO, however, there is a need for SECO specific QA strategies.

Finally, a field that has not been covered in the literature, is the organization of and decision making in SECOs. We recognize the high differentiation in the management models existing SECOs apply that would probably give reasons to why this field is not addressed in the literature. However, we argue that studies on that aspect of SECOs would assist, providing a more complete picture of the field.

6. Conclusion

Software ecosystems is an area that has been gaining in popularity the last five years. The software industry is moving towards software ecosystems, with platforms like Google Android and Apple iOS increasing in popularity, while research has increasing interest in the field, with the fourth year of a dedicated workshop (IWSECO 2012). This article is documenting a systematic literature review held on the field of software ecosystems. The purpose of this work was to provide an overview of the field and identify possible research issues or areas not covered. We found and analyzed 90 relevant papers from a gross total of 420 extracted from a list of scientific libraries. Based on this, we provided an overview of the definition of SECOs as it is defined in the literature including finding patterns in the different definitions provided and list the common main items that consist a SECO. We reported an increase in the research from 2007 to today. Additionally, we classified the research papers according to the result they reported and identified a lack in analytical models and an excess in report papers. Moreover, we defined "SECO architecture" and identified and analyzed the three main components that is consisted of: SECO Software Engineering, SECO Business and Management and SECO Relations. Finally, we examined the intersection of research and industry and found that half of the papers relate to the industry while at the same time most of them are focusing on FOSS SECOs. In conclusion, we identify the field of software ecosystems as a new field of growing importance and potential both in research and industry.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments that greatly improved the quality of this paper.

This work has been partially funded by the Net4Care project within Caretech Innovation (http://www.caretechinnovation.dk/projekter/net4care/).

Appendix A. Literature body

1. Capuruço and Capretz (2010)
2. Popp (2011)
3. Yu and Deng (2011)

4. te Molder et al. (2011)
5. dos Santos and Werner (2011b)
6. van Angeren et al. (2011)
7. Mens and Goeminne (2011)
8. Barbosa and Alves (2011)
9. Alspaugh et al. (2009)
10. Jansen et al. (2009a)
11. Fricker (2009)
12. Campbell and Ahmed (2010)
13. Seichter et al. (2010)
14. Dhungana et al. (2010)
15. Lungu et al. (2010b)
16. van den Berk et al. (2010)
17. Cataldo and Herbsleb (2010)
18. Goeminne and Mens (2010)
19. Bosch (2010a)
19. Bosch (2009)
21. Kazman and Chen (2010)
22. Lungu and Lanza (2010)
23. Pettersson et al. (2010)
24. Scacchi (2010b)
25. Boucharas et al. (2009)
26. Brummermann et al. (2011)
27. Anvaari and Jansen (2010)
28. Hunink et al. (2010)
29. dos Santos and Werner (2010)
30. McGregor (2010)
31. Scacchi (2007a)
32. Krishna and Srinivasa (2011)
33. Alves and Pessôa (2010)
34. Briscoe (2010)
35. Fricker (2010)
36. Scacchi (2010a)
37. Hilkert et al. (2010)
38. Bosch and Bosch-Sijtsema (2010c)
39. Bosch and Bosch-Sijtsema (2010a)
40. Kazman et al. (2012)
41. Schneider et al. (2010)
42. Yu and Woodard (2009)
43. Bosch (2010b)
44. Hanssen (2011)
45. Bosch and Bosch-Sijtsema (2010b)
46. Scacchi (2007b)
47. Lungu et al. (2010a)
48. Brewer and Johnson (2010)
49. Pettersson and Gil (2010)
50. Yu et al. (2008)
51. Lungu et al. (2009)
52. An (2009)
53. Janner et al. (2008)
54. Lungu (2008)
55. Hindle et al. (2010)
56. Jansen et al. (2009b)
57. Yu et al. (2007)
58. Schugerl et al. (2009)
59. Kakola (2010)
60. Ververs et al. (2011)
61. van Angeren et al. (2011)
62. Scholten et al. (2012)
63. Brummermann et al. (2012)
64. Stefanuto et al. (2011)
65. van der Schuur et al. (2011)
66. van Ingen et al. (2011)
67. Weiss (2011)
68. Robbes and Lungu (2011)
69. Schmerl et al. (2011)
70. Yu (2011)
71. dos Santos and Werner (2011a)
72. Idu et al. (2011)
73. Draxler et al. (2011a)
74. Jergensen et al. (2011)
75. Scacchi and Alspaugh (2012)
76. Jansen et al. (2012)
77. Kilamo et al. (2012)
78. Pettersson and Vogel (2012)
79. Kajan et al. (2011)
80. Pérez et al. (2012)
81. Neu et al. (2011)
82. Riis and Schubert (2012)
83. Mizushima and Ikawa (2011)
84. Kabbedijk and Jansen (2011)
85. Draxler and Stevens (2011)
86. Burkard et al. (2012)
87. Viljainen and Kauppinen (2011)
88. Alves et al. (2011)
89. Draxler et al. (2011b)
90. Widjaja and Buxmann (2011)

References

Alspaugh, T., Asuncion, H., Scacchi, W., 2009. The role of software licenses in open architecture ecosystems. In: First International Workshop on Software Ecosystems (IWSECO-2009), Citeseer, pp. 4–18.

Alves, A.M., Pessôa, M., 2010. Brazilian public software: beyond sharing. In: in: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 73–80.

Alves, A.M., Pessoa, M., Salviano, C.F., 2011. Towards a systemic maturity model for public software ecosystems. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (Eds.), Software Process Improvement and Capability Determination, vol. 155 of Communications in Computer and Information Science. Springer, Berlin/Heidelberg, pp. 145–156, 10.1007/978-3-642-21233-8_13.

An, H., 2009. Research on software problems based on ecological angle. In: International Conference on Environmental Science and Information Application Technology 2009 (ESIAT 2009), pp. 11–14.

van Angeren, J., Blijleven, V., Jansen, S., 2011. Relationship intimacy in software ecosystems: a survey of the dutch software industry. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 68–75.

van Angeren, J., Kabbedijk, J., Popp, K.M., 2011. A survey of associate models used within large software ecosystems. In: Third International Workshop on Software Ecosystems (IWSECO-2011), CEUR-WS, pp. 27–39.

Anvaari, M., Jansen, S., 2010. Evaluating architectural openness in mobile software platforms. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 85–92.

Barbosa, O., Alves, C., 2011. A systematic mapping study on software ecosystems. In: Third International Workshop on Software Ecosystems (IWSECO-2011), CEUR-WS, pp. 15–26.

Bass, L., Clements, P., Kazman, R., 2003. Software Architecture in Practice. Addison-Wesley Longman Publishing Co., Inc., Boston.

van den Berk, I., Jansen, S., Luinenburg, L., 2010. Software ecosystems: a software ecosystem strategy assessment model. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 127–134.

Bosch, J., 2009. From software product lines to software ecosystems. In: Proceedings of the 13th International Software Product Line Conference, Carnegie Mellon University, Pittsburgh, PA, USA, pp. 111–119.

Bosch, J., 2010a. Architecture challenges for software ecosystems. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 93–95.

Bosch, J., 2010b. Architecture in the age of compositionality. In: Babar, M., Gorton, I. (Eds.), Software Architecture, volume 6285 of Lecture Notes in Computer Science. Springer Berlin/Heidelberg, pp. 1–4, http://dx.doi.org/10.1007/978-3-642-15114-9_1

Bosch, J., Bosch-Sijtsema, P., 2010a. Coordination between global agile teams: from process to architecture. In: Smite, D., Moe, N.B., Agerfalk, P.J. (Eds.), Agility Across Time and Space. Springer, Berlin/Heidelberg, pp. 217–233, http://dx.doi.org/10.1007/978-3-642-12442-6_15

Bosch, J., Bosch-Sijtsema, P., 2010b. From integration to composition. On the impact of software product lines global development and ecosystems. Journal of Systems and Software 83, 67–76, SI: Top Scholars.

Bosch, J., Bosch-Sijtsema, P.M., 2010c. Softwares product lines, global development and ecosystems: Collaboration in software engineering. In: Mistrik, I., van der Hoek, A., Grundy, J., Whitehead, J. (Eds.), Collaborative Software Engineering.

Springer, Berlin/Heidelberg, pp. 77–92, http://dx.doi.org/10.1007/978-3-642-10294-3_4
Boucharas, V., Jansen, S., Brinkkemper, S., 2009. Formalizing software ecosystem modeling. In: in: Proceedings of the 1st International Workshop on Open Component Ecosystems, ACM, New York, NY, USA, pp. 41–50.
Brewer, R., Johnson, P., 2010. Wattdepot: an open source software ecosystem for enterprise-scale energy data collection storage analysis and visualization. In: in: First IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 91–95.
Briscoe, G., 2010. Complex adaptive digital ecosystems. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 39–46.
Brummermann, H., Keunecke, M., Schmid, K., 2011. Variability issues in the evolution of information system ecosystems. In: Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems, ACM, New York, NY, USA, pp. 159–164.
Brummermann, H., Keunecke, M., Schmid, K., 2012. Formalizing distributed evolution of variability in information system ecosystems. In: Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems, ACM, New York, NY, USA, pp. 11–19.
Burkard, C., Widjaja, T., Buxmann, P., 2012. Software ecosystems. Business and Information Systems Engineering 4, 41–44, http://dx.doi.org/10.1007/s12599-011-0199-8.
Campbell, P.R.J., Ahmed, F., 2010. A three-dimensional view of software ecosystems. In: in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 81–84.
Capuruço, R.A.C., Capretz, L.F., 2010. Integrating recommender information in social ecosystems decisions. In: in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 143–150.
Cataldo, M., Herbsleb, J.D., 2010. Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In: in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 65–72.
Dhungana, D., Groher, I., Schludermann, E., Biffl, S., 2010. Software ecosystems vs natural ecosystems learning from the ingenious mind of nature. In: in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 96–102.
Draxler, S., Jung, A., Boden, A., Stevens, G., 2011a. Workplace warriors: identifying team practices of appropriation in software ecosystems. In: in: Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering, ACM, New York, NY, USA, pp. 57–60.
Draxler, S., Jung, A., Stevens, G., 2011b. Managing software portfolios: a comparative study. In: Costabile, M., Dittrich, Y., Fischer, G., Piccinno, A. (Eds.), End-User Development, vol. 6654 of Lecture Notes in Computer Science. Springer, Berlin/Heidelberg, pp. 337–342, http://dx.doi.org/10.1007/978-3-642-21530-8_36
Draxler, S., Stevens, G., 2011. Supporting the collaborative appropriation of an open software ecosystem. Computer Supported Cooperative Work (CSCW) 20, 403–448, http://dx.doi.org/10.1007/s10606-011-9148-9
Fricker, S., 2009. Specification and analysis of requirements negotiation strategy in software ecosystems. In: in: First International Workshop on Software Ecosystems (IWSECO-2009), Citeseer, pp. 19–33.
Fricker, S., 2010. Requirements value chains: stakeholder management and requirements engineering in software ecosystems. In: Wieringa, R., Persson, A. (Eds.), Requirements Engineering: Foundation for Software Quality, vol. 6182 of Lecture Notes in Computer Science. Springer, Berlin/Heidelberg, pp. 60–66, http://dx.doi.org/10.1007/978-3-642-14192-8_7
Goeminne, M., Mens, T., 2010. A framework for analysing and visualising open source software ecosystems. In: in: Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), ACM, New York, NY, USA, pp. 42–47.
Hansen, K.M., Jonasson, K., Neukirchen, H., 2011. Controversy corner. An empirical study of software architectures' effect on product quality. Journal of Systems and Software 84, 1233–1243.
Hanssen, G.K., 2011. A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. Journal of Systems and Software, http://dx.doi.org/10.1016/j.jss.2011.04.020
den Hartigh, E., Tol, M., Visscher, W., 2006. The health measurement of a business ecosystem. In: in: Proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting.
Hilkert, D., Wolf, C.M., Benlian, A., Hess, T., 2010. The "as-a-service"-paradigm and its implications for the software industry – insights from a comparative case study in crm software ecosystems. In: Aalst, W., Mylopoulos, J., Sadeh, N.M., Shaw, M.J., Szyperski, C., Tyrvainen, P., Jansen, S., Cusumano, M.A. (Eds.), Software Business, vol. 51 of Lecture Notes in Business Information Processing. Springer, Berlin/Heidelberg, pp. 125–137, http://dx.doi.org/10.1007/978-3-642-13633-7_11
Hindle, A., Herraiz, I., Shihab, E., Jiang, Z.M., 2010. Mining challenge 2010: Freebsd gnome desktop and debian/ubuntu. In: in: 7th IEEE Working Conference on Mining Software Repositories (MSR), pp. 82–85.
Hunink, I., van Erk, R., Jansen, S., Brinkkemper, S., 2010. Industry taxonomy engineering: the case of the european software ecosystem. In: in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 111–118.
Iansiti, M., Levien, R., 2004a. The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy Innovation and Sustainability. Harvard Business Press, Boston.
Iansiti, M., Levien, R., 2004b. Strategy as ecology. Harvard Business Review 82, 68–81.
Idu, A., van de Zande, T., Jansen, S., 2011. Multi-homing in the apple ecosystem: why and how developers target multiple apple app stores. In: in: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 122–128.
van Ingen, K., van Ommen, J., Jansen, S., 2011. Improving activity in communities of practice through software release management. In: in: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 94–98.
Janner, T., Schroth, C., Schmid, B., 2008. Modelling service systems for collaborative innovation in the enterprise software industry – the st. gallen media reference model applied. In: in: IEEE International Conference on Services Computing 2008 (SCC'08), pp. 145–152.
Jansen, S., Brinkkemper, S., Finkelstein, A., 2009a. Business network management as a survival strategy: a tale of two software ecosystems. In: in: First International Workshop on Software Ecosystems (IWSECO-2009), Citeseer, pp. 34–48.
Jansen, S., Brinkkemper, S., Souer, J., Luinenburg, L., 2012. Shades of gray: opening up a software producing organization with the open software enterprise model. Journal of Systems and Software 85, 1495–1510, Software Ecosystems.
Jansen, S., Finkelstein, A., Brinkkemper, S., 2009b. A sense of community: A research agenda for software ecosystems. In: in: 31st International Conference on Software Engineering – Companion Volume, 2009. ICSE-Companion 2009, pp. 187–190.
Jergensen, C., Sarma, A., Wagstrom, P., 2011. The onion patch: migration in open source ecosystems. In: in: Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering, ACM, New York, NY, USA, pp. 70–80.
Kabbedijk, J., Jansen, S., 2011. Steering insight: An exploration of the ruby software ecosystem. In: Regnell, B., Weerd, I., Troyer, O., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C. (Eds.), Software Business, vol. 80 of Lecture Notes in Business Information Processing. Springer, Berlin/Heidelberg, pp. 44–55, http://dx.doi.org/10.1007/978-3-642-21544-5_5
Kajan, E., Lazic, L., Maamar, Z., 2011. Software engineering framework for digital service-oriented ecosystem. In: in: 19th Telecommunications Forum (TELFOR), pp. 1320–1323.
Kakola, T., 2010. Standards initiatives for software product line engineering and management within the international organization for standardization. In: in: 43rd Hawaii International Conference on System Sciences (HICSS), 2010, pp. 1–10.
Kazman, R., Chen, H.M., 2010. The metropolis model and its implications for the engineering of software ecosystems. In: in: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, ACM, New York, NY, USA, pp. 187–190.
Kazman, R., Gagliardi, M., Wood, W., 2012. Scaling up software architecture analysis. Journal of Systems and Software 85, 1511–1519, Software Ecosystems.
Kilamo, T., Hammouda, I., Mikkonen, T., Aaltonen, T., 2012. From proprietary to open source-growing an open source ecosystem. Journal of Systems and Software 85, 1467–1478.
Kitchenham, B., Charters, S., 2007. Guidelines for performing systematic literature reviews in software engineering. Engineering 2.
Krishna, R.P.M., Srinivasa, K.G., 2011. Analysis of projects and volunteer participation in large scale free and open source software ecosystem. SIGSOFT Software Engineering Notes 36, 1–5.
Lungu, M., 2008. Towards reverse engineering software ecosystems. In: in: IEEE International Conference on Software Maintenance, ICSM 2008, pp. 428–431.
Lungu, M., Lanza, M., 2010. The small project observatory: a tool for reverse engineering software ecosystems. In: in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, ACM, New York, NY, USA, pp. 289–292.
Lungu, M., Lanza, M., Gîrba, T., Robbes, R., 2010a. The small project observatory: visualizing software ecosystems. Science of Computer Programming 75, 264–275, Experimental Software and Toolkits (EST 3): A special issue of the Workshop on Academic Software Development Tools and Techniques (WASDeTT 2008).
Lungu, M., Malnati, J., Lanza, M., 2009. Visualizing gnome with the small project observatory. In: in: Proceedings of the 6th IEEE International Working Conference on Mining Software Repositories, 2009 (MSR'09), pp. 103–106.
Lungu, M., Robbes, R., Lanza, M., 2010b. Recovering inter-project dependencies in software ecosystems. In: in: Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, ACM, New York, NY, USA, pp. 309–312.
McGregor, J.D., 2010. A method for analyzing software product line ecosystems. In: in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 73–80.
Mens, T., Goeminne, M., 2011. Analysing the evolution of social aspects of open source software ecosystems. In: in: Third International Workshop on Software Ecosystems (IWSECO-2011), CEUR-WS, pp. 1–14.
Messerschmitt, D., Szyperski, C., 2005. Software Ecosystem: Understanding An Indispensable Technology and Industry. MIT Press Books 1, London, England.
Mizushima, K., Ikawa, Y., 2011. A structure of co-creation in an open source software ecosystem: a case study of the eclipse community. In: in: 2011 Proceedings of

PICMET'11, Technology Management in the Energy Smart World (PICMET), pp. 1–8.
te Molder, J., van Lier, B., Jansen, S., 2011. Clopenness of systems: The interwoven nature of ecosystems. In: in: Third International Workshop on Software Ecosystems (IWSECO-2011), CEUR-WS, pp. 52–64.
Neu, S., Lanza, M., Hattori, L., D'Ambros, M., 2011. Telling stories about gnome with complicity. In: in: 2011 6th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT), pp. 1–8.
Pettersson, O., Gil, D., 2010. On the issue of reusability and adaptability in m-learning systems. In: in: 2010 6th IEEE International Conference on Wireless, Mobile and Ubiquitous Technologies in Education (WMUTE), pp. 161–165.
Pettersson, O., Svensson, M., Gil, D., Andersson, J., Milrad, M., 2010. On the role of software process modeling in software ecosystem design. In: in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 103–110.
Pettersson, O., Vogel, B., 2012. Reusability and interoperability in mobile learning: a study of current practices. In: in: 2012 IEEE Seventh International Conference onWireless, Mobile and Ubiquitous Technology in Education (WMUTE), pp. 306–310.
Pérez, J., Deshayes, R., Goeminne, M., Mens, T., 2012. Seconda: software ecosystem analysis dashboard. In: in: 16th European Conference on Software Maintenance and Reengineering (CSMR), pp. 527–530.
Popp, K., 2010. Goals of software vendors for partner ecosystems – a practitioner's view. Software Business, 181–186.
Popp, K.M., 2011. Hybrid revenue models of software companies and their relationship to hybrid business models. In: in: Third International Workshop on Software Ecosystems (IWSECO-2011), CEUR-WS, pp. 77–88.
Riis, P., Schubert, P., 2012. Upgrading to a new version of an erp system: a multilevel analysis of influencing factors in a software ecosystem. In: in: 45th Hawaii International Conference on System Science (HICSS), pp. 4709–4718.
Robbes, R., Lungu, M., 2011. A study of ripple effects in software ecosystems (nier track). In: in: Proceedings of the 33rd International Conference on Software Engineering, ACM, New York, NY, USA, pp. 904–907.
dos Santos, R.P., Werner, C., 2011a. Treating business dimension in software ecosystems. In: in: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 197–201.
dos Santos, R.P., Werner, C.M.L., 2010. Revisiting the concept of components in software engineering from a software ecosystem perspective. In: in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 135–142.
dos Santos, R.P., Werner, C.M.L., 2011b. A proposal for software ecosystem engineering. In: in: Third International Workshop on Software Ecosystems (IWSECO-2011), CEUR-WS, pp. 40–51.
Scacchi, W., 2007a. Free/open source software development: recent research results and emerging opportunities. In: in: The 6th Joint Meeting on European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering: Companion Papers, ACM, New York, NY, USA, pp. 459–468.
Scacchi, W., 2007b. Free/open source software development: recent research results and methods. In: Zelkowitz, M.V. (Ed.), Architectural Issues, Elsevier, vol. 69 of Advances in Computers. , pp. 243–295.
Scacchi, W., 2010a. Collaboration practices and affordances in free/open source software development. In: Mistrík, I., van der Hoek, A., Grundy, J., Whitehead, J. (Eds.), Collaborative Software Engineering. Springer, Berlin/Heidelberg, pp. 307–327, http://dx.doi.org/10.1007/978-3-642-10294-3_15
Scacchi, W., 2010b. The future of research in free/open source software development. In: in: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, ACM, New York, NY, USA, pp. 315–320.
Scacchi, W., Alspaugh, T.A., 2012. Understanding the role of licenses and evolution in open architecture software ecosystems. Journal of Systems and Software 85, 1479–1494.
Schmerl, B., Garlan, D., Dwivedi, V., Bigrigg, M.W., Carley, K.M., 2011. Sorascs: a case study in soa-based platform design for socio-cultural analysis. In: Proceedings of the 33rd International Conference on Software Engineering, ACM, New York, NY, USA, pp. 643–652.
Schneider, K., Meyer, S., Peters, M., Schliephacke, F., Mörschbach, J., Aguirre, L., 2010. Feedback in context: supporting the evolution of it-ecosystems. In: Ali Babar, M., Vierimaa, M., Oivo, M. (Eds.), Product-Focused Software Process Improvement, vol. 6156 of Lecture Notes in Computer Science. Springer, Berlin/Heidelberg, pp. 191–205, 10.1007/978-3-642-13792-1_16.
Scholten, U., Fischer, R., Zirpins, C., 2012. The dynamic network notation: harnessing network effects in paas-ecosystems. In: Proceedings of the Fourth Annual Workshop on Simplifying Complex Networks for Practitioners, ACM, New York, NY, USA, pp. 25–30.
Schugerl, P., Rilling, J., Witte, R., Charland, P., 2009. A quality perspective of software evolvability using semantic analysis. In: IEEE International Conference on Semantic Computing, ICSC'09, pp. 420–427.
van der Schuur, H., Jansen, S., Brinkkemper, S., 2011. The power of propagation: on the role of software operation knowledge within software ecosystems. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 76–84.
Seichter, D., Dhungana, D., Pleuss, A., Hauptmann, B., 2010. Knowledge management in software ecosystems: software artefacts as first-class citizens. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, pp. 119–126.
Shaw, M., 2003. Writing good software engineering research papers. In: Minitutorial for Proc ICSE"03.
Stefanuto, G., Spiess, M., Alves, A.M., Castro, P.F.D., 2011. Quality in software digital ecosystems the users perceptions. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 85–88.
Ververs, E., van Bommel, R., Jansen, S., 2011. Influences on developer participation in the debian software ecosystem. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, New York, NY, USA, pp. 89–93.
Viljainen, M., Kauppinen, M., 2011. Software ecosystems: a set of management practices for platform integrators in the telecom industry. In: Regnell, B., Weerd, I., Troyer, O., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M.J., Szyperski, C. (Eds.), Software Business, vol. 80 of Lecture Notes in Business Information Processing. Springer, Berlin/Heidelberg, pp. 32–43, 10.1007/978-3-642-21544-5_4.
Weiss, M., 2011. Economics of collectives. In: Proceedings of the 15th International Software Product Line Conference, vol. 2, ACM, New York, NY, USA, pp. 39:1–39:8.
Widjaja, T., Buxmann, P., 2011. Compatibility of software platforms. In: Heinzl, A., Buxmann, P., Wendt, O., Niche Weitzel, T. (Eds.), Theory-Guided Modeling and Empiricism in Information Systems Research. Physica-Verlag HD, Berlin Heidelberg, Germany, pp. 15–41, http://dx.doi.org/10.1007/978-3-7908-2781-1_2
Yu, E., Deng, S., 2011. Understanding software ecosystems: A strategic modeling approach. In: Third International Workshop on Software Ecosystems (IWSECO-2011), CEUR-WS, pp. 65–76.
Yu, L., 2011. Coevolution of information ecosystems: a study of the statistical relations among the growth rates of hardware, system software, and application software. SIGSOFT Software Engineering Notes 36, 1–5.
Yu, L., Ramaswamy, S., Bush, J., 2007. Software evolvability: an ecosystem point of view. In: in: Third International IEEE Workshop on Software Evolvability, pp. 75–80.
Yu, L., Ramaswamy, S., Bush, J., 2008. Symbiosis and software evolvability. IT Professional 10, 56–62.
Yu, S., Woodard, C., 2009. Innovation in the programmable web: characterizing the mashup ecosystem. In: Feuerlicht, G., Lamersdorf, W. (Eds.), Service-Oriented Computing – ICSOC 2008 Workshops, vol. 5472 of Lecture Notes in Computer Science. Springer, Berlin/Heidelberg, pp. 136–147, http://dx.doi.org/10.1007/978-3-642-01247-1_13.

Konstantinos Manikas is a PhD scholar at the Department of Computer Science of Copenhagen University. His main research areas are software architecture and software ecosystems with interest in telemedicince and healthcare IT.

Klaus Marius Hansen is a professor of Software Development at the University of Copenhagen. He received a Ph.D. degree in Computer Science from Aarhus University in 2002 and his research focuses on software technology and use in particular in relation to pervasive and dependable computing.