



---

# Software Ecosystems

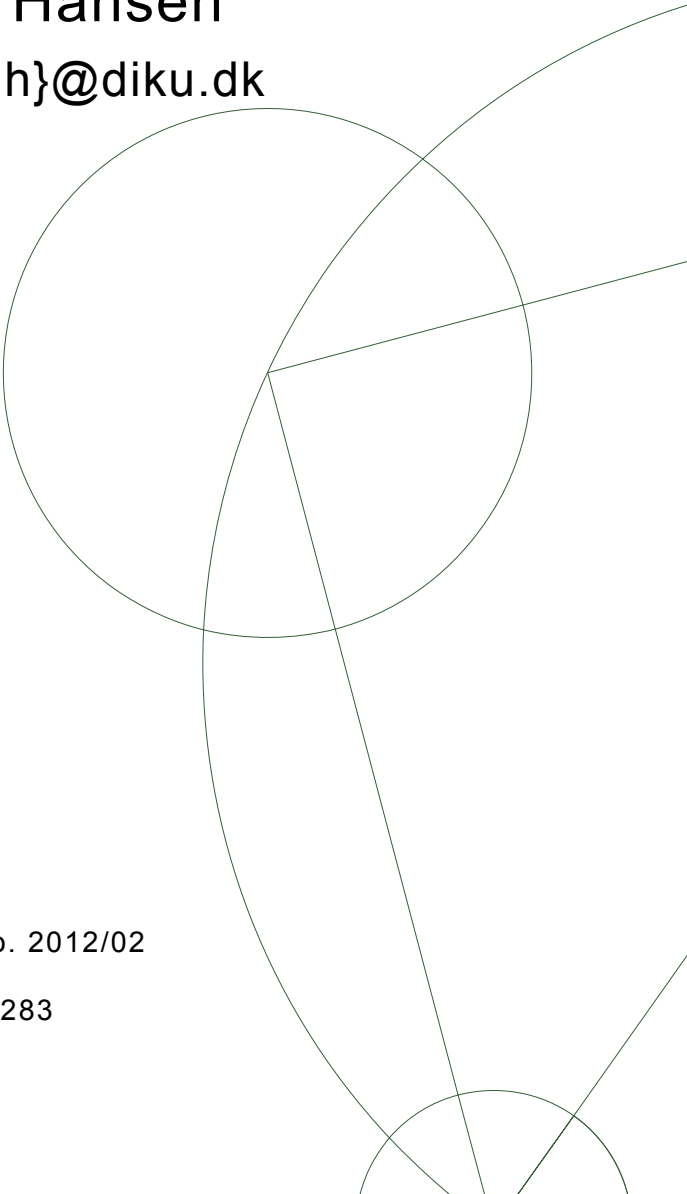
A Systematic Literature Review

Konstantinos Manikas and  
Klaus Marius Hansen

{kmanikas,klausmh}@diku.dk

Technical Report no. 2012/02

ISSN: 0107-8283



## Abstract

Software ecosystems are sets of software solutions functioning as a unit, enabling actors to automate activities and transactions. Arguably, software ecosystems are gaining importance with the advent of, e.g., the Google Android, Apple iOS, 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 59 papers on the subject taken from a gross collection of 212. Our main conclusions are that while research on software ecosystems is increasing there is a) 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.

**Keywords:** Software ecosystems, software ecosystem, systematic literature review

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Report structure . . . . .	3
<b>2</b>	<b>Review protocol</b>	<b>3</b>
2.1	Research questions . . . . .	4
2.2	Data collection strategy . . . . .	6
2.3	Data extraction . . . . .	7
2.4	Data synthesis . . . . .	7
2.5	Dissemination . . . . .	8
<b>3</b>	<b>Literature collection</b>	<b>8</b>
<b>4</b>	<b>Analysis</b>	<b>9</b>
4.1	Yearly activity . . . . .	11
4.2	Research result . . . . .	11
4.3	SECO Architecture . . . . .	12
4.4	Connection with industry . . . . .	14
<b>5</b>	<b>Conclusion</b>	<b>15</b>

# 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 internal and external to the organization developing the platform [1, 2]. In this setting, software engineering is spread outside the traditional borders of software companies to a group of companies, private persons, or 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, co-exist 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<sup>1</sup>. This is a parallel to the natural ecosystems where the different members of the ecosystems (e.g., the plants, animals, or insects) are part of a food chain 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<sup>2</sup>.

While software ecosystems are thus arguably gaining importance, research in software ecosystems is in its infancy, starting in 2005 with [3] and now with a dedicated workshop in its third year<sup>3</sup>. Our own literature search (see Section 3) revealed a gross list of 212 published papers on software ecosystems. However, until now there has been no systematic review 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 et al. [4]. 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 efficiently. This work is intended to function as a snapshot of the research in the field by i) identifying and analyzing the different definitions of the SECO, 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 what extent the research is connected to SECO industry.

## 1.1 Report structure

The rest of this report is organized as following: Section 2 specifies the review protocol, Section 3 reports on the extracted data, Section 4 analyses extracted data, and Section 5 concludes.

# 2 Review protocol

The applied review protocol is based on the approach of [4]. The establishment of the review protocol is necessary to ensure that the literature

---

<sup>1</sup><http://developer.apple.com/programs/ios/distribute.html>

<sup>2</sup><http://www.slideshare.net/CMSummit/ms-internet-trends060710final>

<sup>3</sup><http://www.softwareecosystems.org/workshop/>

review is systematic and to minimize the researcher bias. As such, the literature review is focused on a set of research questions that consist 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. The data collection strategy defines the details of the literature collection in terms of search queries, scientific libraries and inclusion-exclusion criteria. The data extraction process, defines what information should be collected from the relevant literature in order to address the initial research questions. Finally, the data synthesis section addresses the mapping of the extracted data with the research questions.

## 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. We intend address the following research questions:

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

In order to be able to analyze the field of SECO, 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 will 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 will be able to identify possible trends in the research invested in the field of SECO. 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 the field of SECO is changing within time.

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

After having defined the term SECO, a question that rises 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 [5] has been chosen as a starting point. 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 in [6], Kazman et. al are analyzing a series of traditional software design and software architecture principles and methods in the perspective of the SECO or software-intensive ecosystems as they are mentioned. This results in some new or adapted methods for the software design and architecture of these software-intensive ecosystems.

**Qualitative or descriptive model** A model based on qualitative analysis of data or well argumentation of existing cases. The paper provides 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 case study of two SECOs presented in [7].

**Empirical model** A model derived from the quantitative data collection of the problem area. A paper of this category would base some analysis or predicting model on the study of existing cases. For example the paper of [8] where they are extracting information from open source systems to assess the evolvability of software in terms of software symbiosis, co-evolution of software, or Darwinism, the fittest survives.

**Analytic model** A model using the automatic or mathematical manipulation of the specific problem. For example the prediction of recommendation and interaction between the members of a social ecosystem based on a mathematical analysis of the member relationships proposed in [9].

**Tool or notation** A tool 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 [10]

**Specific solution, prototype, answer, or judgment** A complete solution, evaluation of a theory, comparison of different theories based on a software engineering problem. The result is addressing a specific problem. An example would be [11] where they are addressing the reusability and adaptability issues in mobile learning systems

**Report** Knowledge and experience obtained, rules of thumb or checklists but not systematic enough to be a descriptive model. For example, the report of the hybrid business and revenue models that software companies can have reported in [12].

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 [13, 14]. In relation to this, we will analyze the extent to which SECO literature stresses software architecture. We will 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 [13]:

“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 both systems, system components, and actors. Relationships then include software architecture-related relationships as well of actor-related relationships such the adaptation of new members.

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 the research are in the field of software ecosystems: research benefits from realism of problems when connected to the industry while industry eventually may become more innovative and efficient when connected to research.

We investigate how close the research world is with the industry by examining how much of the literature has focused on real-world SECOs. We accept that a report has focus on a real SECO when it either presents a real SECO as an object of study or using the data from the study of one to support a claim or result. For example, this could be a paper that is presenting a new structure model of SECOs that has already been applied in a SECO. Another example could be deducting information of the external members of an ecosystem by studying the relationships between the members of one or more real SECOs. However, we did 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 Data collection strategy

The strategy for collecting relevant literature was 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<sup>4</sup>
2. IEEE Explore<sup>5</sup>
3. Springer Verlags' digital library, SpringerLink<sup>6</sup>
4. ScienceDirect<sup>7</sup>. An online collection of published scientific research operated by the publisher Elsevier.
5. Thomson Reuters' Web of Science<sup>8</sup>. An online academic citation index.

The literature extraction consisted of two separate keyword searches with the search terms “software ecosystem” and “software ecosystems” in the libraries above. The search query was intentionally kept simple so we could extract the maximum number of papers containing the terms software ecosystem(s).

With respect to ii), we included articles from all three International Workshops on Software Ecosystems (IWSECO): IWSECO 2009, IWSECO 2010 and IWSECO 2011.

The included literature was articles and journals that contained the words “software ecosystem(s)” in the title, keywords or abstract. For each article extracted from the libraries, we would search for the terms “software ecosystem(s)” in the pre-mentioned fields. Articles that came up on the search results in the libraries because “software ecosystem(s)” was in their text body and not in any of the title, abstract or keywords

---

<sup>4</sup><http://dl.acm.org/>

<sup>5</sup><http://ieeexplore.ieee.org>

<sup>6</sup><http://www.springerlink.com/>

<sup>7</sup><http://www.sciencedirect.com/>

<sup>8</sup><http://apps.webofknowledge.com/>

fields, we searched for only the word “ecosystem(s)” in the same fields in case they would be mentioning something similar ex. “software-intensive ecosystems”

In conclusion, this study included papers from conferences, workshops or journals that were written in english language, claim to report research and contain the terms “software ecosystem(s)” or something referring to software before the words “ecosystem(s)” and be more than one page long. We excluded from the studied literature any books, conference keynotes and extended abstracts.

### 2.3 Data extraction

After compiling the list with the papers to be included in the literature review, we would read each paper and collect information concerning the paper. The collected information from all the relevant literature would be the data that drive the results of this work. Therefore the collected information is directly related to the research questions and formed accordingly:

- A concise *summary* of the paper giving somewhat different information from the abstract. The description summarizes what the authors report and may provide additional comments on the paper.
- The paper *publication year*.
- The *definition* or *definitions* of software ecosystem used and, in case of quoting a definition from another paper, the source of definition.
- The *classification of the result* according to the list presented in the research question 3.
- Whether *architectural considerations* of SECOs were addressed and how were they addressed.
- The use of *real SECOs* as an object of study in the paper. The real SECO should be either the main focus of the paper or used as a use case to argue a claim. In any case the SECO mentioned should have been researched from some perspective. The section would include the name of the ecosystem and relevant information on the ecosystem.

### 2.4 Data synthesis

Having extracted the necessary information per paper, we had to organize our data in a way that could be easier manipulated. Therefore we created an additional dataset where the data were organized to address the research questions. For each research question, we collected the following information:

- RQ 1** We listed the collection of definitions and identified similarities among the definitions or definitions referring to other papers. This would help us to discover patterns or similarities in the way the different articles define the field of SECO.
- RQ 2** We listed the publication year of all the papers and examined on if it shows any pattern in the number of articles per year: increasing, decreasing or stable.
- RQ 3** Each paper was classified on the previously mentioned categories. Listing the total papers on each category, gives us an overview on what kind of papers has the research community been focusing more.

- RQ 4** The number of articles focusing on the SECO architecture gave a first view on how important architecture is for the research community. For each of these papers, we collected information on what perspective of the SECO architecture they are addressing.
- RQ 5** A list of the articles using a real SECO in their study including the name of the SECO and any additional information about the SECO we would find relevant. The additional information could be used for making conclusions on if there are common characteristics on the SECOs used in the papers.

## 2.5 Dissemination

The findings of this work would initially be a technical report and then a chapter in a PhD thesis focusing on software ecosystems. Additionally, as it might be of interest to the research community, we will aim at publishing it in a scientific journal or conference.

## 3 Literature collection

We divided the data extraction process in two steps: i) the *Identification of relevant papers* and ii) the *paper data extraction*.

The identification of relevant papers was done in three passes, as we show in table 1. The first pass was the scientific library keyword search. The keyword search in the list of scientific libraries, as described in chapter 2.2, gave a total of 212 papers. The search was conducted by entering the search queries in the generic search field.

In the second pass, we searched for the terms “software ecosystem(s)” in their title, abstract or keywords in the 212 papers that resulted from the previous step. This step gave 87 papers. The excluded papers from this step, that is the 127 papers that did not contain the keywords in any of the title, abstract or keyword fields, were searched for only the words “ecosystem(s)” in the same fields and evaluated what would precede the searched word in case we would find terms like “digital ecosystems” or “software-intensive ecosystems”. This provided one paper that was added to the 87 included papers. We explain the high number of papers rejected as not relevant from this step due 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 did not recognize the quotation marks in “software ecosystem” or “software ecosystems” so it would retrieve results that the words were not consequent to each other but in different locations in the texts, therefore there were many papers not related to software engineer.

Finally, we conducted a manual evaluation of the papers that resulted from the previous step (88 in total) by reading the abstract and conclusion and evaluating if they were related to the field of SECO. This resulted in 48 papers relevant to the field of SECO. In the 48 relevant papers, we added 11 papers from the three IWSECO workshops (IWSECO 2009, IWSECO 2010 and IWSECO 2011) that were not already included.

In total, from the initial 212 papers, we selected 59 as reporting research relevant to the field of SECO.

During the data extraction process, we read the papers found relevant and extracted the information needed to address the research questions as described in section 2.4. The information extraction was of the form

Table 1: The 3-pass process for defining the relevant literature.

Pass Nr.	Input(papers)	Description	Resulting papers
1		Scientific Library Search	212
2.a	212 from (1)	Title, abstract, and keywords search for “software ecosystem(s)”	87
2.b	127 rejected from (2.a)	Title, abstract, and keywords search for “ecosystem(s)” and evaluating preceding words	1
3.a	88 from (2.a)+(2.b)	Manual abstract and conclusion evaluation	48
3.b		Include the IWSECO papers not included	11
	(3.a)+(3.b)	Total Number or papers found relevant	59

of descriptive text enclosed by identifying labels for automated sorting. Initially, we ordered the information per paper and then grouped up the information according to the research question they were addressing as mentioned in section 2.3. This would ease the information manipulation and possible pattern identification.

## 4 Analysis

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 (29 out of the total of 59) 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 was not the general field of SECO. For example, Bosch [15] is not providing any definition, but he is referring back to his own work [1] where he provides a definition and more detailed analysis of the field. On the other hand, Popp [12] 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 one third 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, 22 in total (including the two mentioned previously), are defining the field by using one or more definitions from the existing literature. When we analyzed the definitions we found out that we can group the quoted definitions in four groups according to the source of the definition:

**Messerschmitt et al. [3]** This 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. [3]

**Jansen et al. [16]** The articles that quote Jansen 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 underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts. [16]

**Bosch et al. [1, 17, 18]** Bosch provides two definitions in his 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. [1]

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. [17, 18]

**Lungu [19]** Finally, Lungu is presenting a different definition of the SECOs that is adapted by a number of papers:

A software ecosystem is a collection of software projects which are developed and evolve together in the same environment. [19]

In table 2 we show the different groupings and the papers belonging to each group.

Not surprisingly, if we look at the definitions we can see that all of them 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 take on the definitions, we note that Messerschmitt and Lungu have a pure technical perspective by talking about software and its symbiosis/co-existence, while Bosch and Jansen 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 and Jansen, which are refereed by the majority of the papers that provide a definition for SECO (66%), we can identify three main elements in their definitions:

**Common Software** The software appears either as a “common technological platform” [16], “software solutions” [1] or “software platform” [17, 18]

**Business** This is expressed as either “a set of business” [16], “business ecosystem” [1], a community of users that have needs to be satisfied [17, 18]. In this element the term *Business* is given with a wider sense of the term than the profit or revenue models. This element also includes possible benefits other than financial revenues. For

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

Definition	Papers	Total
Not Available	[15, 9, 18, 20, 21, 12, 22, 23, 24, 25, 26, 11, 27, 28, 29, 10, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]	29
Jansen et al.	[43, 2, 44, 45, 46, 7, 47, 48, 49, 50, 51]	11
Own	[17, 52, 53, 54, 16, 1, 19, 55, 49, 56]	10
Bosch et al.	[57, 44, 58, 59, 47, 60, 61, 49, 6]	9
Messerschmitt et al.	[8, 17, 7, 62]	4
Lungu et al.	[63, 64, 55]	3

example, 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 ” [16], “actors in the associated social ecosystem” [1], “community of domain experts” and “community of users” [17, 18]

Combining the definitions, we can conclude that a software ecosystem consists of a group of actors, one or more business models that serve these actors in a possible wider sense than direct revenues, one or more software platforms that the business models are built upon and the relationships of the actors and business models.

## 4.1 Yearly activity

Another point of study in this work, was the analysis of the year of publication. When we order the papers according to their publication year, we could identify an increasing pattern, as can be seen on table 3. The literature on SECO starts in 2007 (although [3] 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 – are providing an equal number of papers. However, an increase appears in 2009 and continues to 2010. The year 2011 is not taken into consideration, as it was not possible to have the full number of papers for that year during the study.

One of the possible reasons that influenced this increase can be the establishment of the International Workshop on Software Ecosystems (IWSECO) in 2009 that has been held every year every since. However, this alone would not justify 32 articles for 2010. We can, therefore, note that there is a early increase of interest in the research community towards the field of SECO.

## 4.2 Research result

As noted in the research question 3, it is of interest to examine what kind of result the papers are reporting. We have classified the papers in the categories listed in research question 3 and can be seen in table 4. As it can be seen from the table, the majority of the articles fall under the Report category. This means that these articles have as contribution knowledge and experience obtained, rules of thumb or checklists or

Table 3: Papers published per year

Year	Papers	Total
2007	[34, 33, 62]	3
2008	[8, 31, 10]	3
2009	[24, 44, 25, 1, 50, 30, 29, 28, 16, 54]	10
2010	[9, 49, 58, 60, 64, 45, 61, 55, 15, 36, 35, 59, 42, 51, 46, 48, 53, 38, 56, 37, 32, 7, 17, 20, 57, 21, 18, 19, 26, 11, 40, 27]	32
2011	[12, 43, 2, 22, 23, 63, 52, 39, 41, 6, 47]	11

interesting observations but their findings are either not generic enough to allow their adaption 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 [60] that compares 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 used data is not systematic enough to be included in the Qualitative model.

Looking at the percentage list, we note that the category with the most papers after Report is the Procedure or Technique. This category includes papers that report an implementable technique to solve a specific task. For example the paper by Fricker [25] 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 field in the research, with the first papers appearing in 2007. This implies that there is an amount of research resource spent in defining the field and its limits. For example the papers [44, 12] that are providing definitions for SECO structure and SECO business models. In addition, as it is proven in section 4.4, 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 a wide field that collects the interest of several traditional research areas. In connections to this, there have been several papers focusing on some specific aspect of the field providing specific and implementable techniques. This explains the high percentage in the Procedure or Technique category. Although this would also imply a high percentage in the Tool or Notation category, there were not as many papers reporting a new tool as in the Procedure or Technique and that is mainly due to the fact that since the field is so new, the majority of the research focusing on a specific aspect would try to define a framework or technique that can be the base of a future tool.

### 4.3 SECO Architecture

To address the research question 4, 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:

**SECO Management** This group reports on the aspects of the ecosys-

Table 4: The papers grouped according to the result groups.

Definition	Papers	% of total
Report	[15, 21, 12, 23, 44, 52, 27, 28, 16, 29, 32, 33, 34, 1, 37, 49, 39, 41, 42, 60, 48, 38, 61]	37
Procedure or Technique	[6, 57, 22, 25, 53, 31, 45, 58, 46, 55, 60, 61]	19
Qualitative or Descriptive Model	[17, 20, 43, 2, 63, 36, 7, 48, 56]	14
Tool or Notation	[24, 26, 54, 10, 64, 35, 19, 50]	13
Empirical Model	[18, 8, 47, 62, 40, 51]	10
Specific Solution	[11, 59, 38]	5
Analytic Model	[9, 30]	3

tems that are related to the management and organization. Papers belonging to this group might have proposals for business models for SECO, considerations on the SECO health, evaluation of how open the SECO platform is, checklists on how to create a SECO for a specific domain or how to go towards a SECO from different organizational model.

**SECO Actor Relationships** This group focuses on the relationships between the actors involved in a SECO. Papers included in this category might suggest measurements of how open or closed the actor incorporation might be in a SECO, how is the communication established, frameworks for SECO requirement elicitation, external developers' quality assurance, knowledge transfer within the actors of the SECO.

**SECO Software Engineering** This groups represents the perspectives of software engineering and software architecture that are adapted to the field of SECO. Papers falling under this category might be papers discussing the importance of interfaces, reusability, different software architectural patterns, or development in remote teams in the SECO scene.

We formulated the groups according to the context of the papers of study. For each of the papers, we deducted the main focus of the paper and by logically grouping, we ended up in the three pre-mentioned groups. We also note that the three architectural groups can be paralleled with the three elements describing the definition of SECO defined in the definition analysis section. In this sense, the *SECO SE* could address the *Common Software* element in the definition, the *SECO Actor Relationships* could be focusing in the *Connecting Relationships* and the *SECO Management* could be closer to the *Business* element in the definition. We argue that these are the main perspectives that consist a SECO since by removing any of the three there is no SECO. If there is no software engineering perspective the ecosystem is not a *software* ecosystem. In the lack of various actors and the relationship among them, the software ecosystem is lacking the ecosystem perspective and becomes closer to traditional in-house development or product lines. Finally, without the management perspective that could be anything from a traditional management group taking decisions to a set or rules for all the actors to commonly participate the SECO would not be able to coordinate and work aligned as a healthy

Table 5: The articles according to the SECO Architecture groups.

SECO Architecture Group	Papers	% of Total
SECO Management	[12, 2, 22, 44, 25, 54, 16, 45, 1, 59, 7, 47, 37, 60, 38, 49, 50, 51]	43
SECO Actor Relationship	[43, 2, 23, 53, 58, 34, 36, 38, 61]	21
SECO SE	[6, 15, 18, 17, 20, 21, 11, 54, 32, 33, 64, 35, 19, 48, 61, 62]	36

ecosystem.

Table 5 shows the distribution of papers according to the architectural perspective. We notice that the majority of the papers have been focusing their research in the SECO Management group. We explain this mainly based on the fact that the field of SECO is new in the research, therefore the most effort is spent in analyzing and understanding the organizational and business perspectives. Additionally, it is harder to make technical contributions in a SECO than a traditional product line or single system because of the first being traditionally more complicated than the second. Not surprisingly, the papers focusing on applying traditional software engineering perspectives is relatively close to the amount of papers in the management group while the papers of actors' relationships groups are distinctively lower. We could explain this by arguing that analyzing the relationships would possibly require the study of an existing SECO. This could pose an inhibition to the study because of the following reasons: i) the immaturity of the field results in not so many industrial SECOs and ii) the relationships between the actors or the actors and the platform might not be easy to elicit because of the confidentiality of the data.

#### 4.4 Connection with industry

From the research questions that are mentioned in the beginning of this report, question 5 is investigating the use of real SECOs in the research. The purpose is to give a view on how close the connection of the research is to the industry. In the data collection process, we have compiled a list with all the papers that are using an existing SECO in their research either as a case study or a use case. Analyzing this list, we end up in the results that can be seen in table 6. Going through the results, we notice that a part of the papers (39%) are using an existing SECO in their research. It could be argued that it may lead to higher 'external' validity of the results. When examining what kind of SECOs these papers are studying, initially we notice that there is only a small overlap of papers using the same SECO, a fact that implies that several different SECOs have been studied in different papers. We also notice, however, that there is a high percentage of papers (17% of the total number of paper, ie., almost half of the papers using a real SECO) that are studying a SECO of Free or Open Source Software (FOSS). This indicates that the research community is aware of the importance of the study of industrial data. However it is not easy to study existing SECOs due to the fact that many SECOs are closed communities and it is hard to get access to information. Therefore, the use of FOSS SECOs is easier as the possible information that characterize the SECO is open to the public in most of the cases. Additionally, we

Table 6: The papers using a real SECO

SECO type	Papers	% of total
Real SECO (Excl. FOSS)	[18, 57, 12, 44, 53, 10, 31, 45, 7, 47, 38, 50, 39]	22
FOSS SECO	[23, 63, 26, 29, 8, 30, 34, 55, 62, 51]	17
No SECO	[6, 15, 9, 17, 20, 21, 43, 2, 22, 24, 25, 52, 11, 27, 54, 28, 16, 58, 32, 33, 1, 59, 64, 35, 19, 36, 46, 37, 60, 48, 61, 49, 56, 40, 41, 42]	61

notice that some of the studies that use FOSS can be more technical as they are able to get deep technical details that are hard to get access to in a commercial SECO like, for example, the number of developers making change to a class during a release.

However, using a FOSS SECO instead of a commercial one for studying problems related to the *SECO Management* or *SECO Actor Relationships* fields is a decision that might influence the results of the study, depending on the research focus as there are several differences in these two types of SECO. One of the biggest differences is the actor involvement. In a FOSS SECO, the involved actors usually do not have direct financial benefits from this involvement. This can influence the SECO in being more open to new actors, affect the actor communication and as a result the way the SECO is managed. In a commercial SECO, the involved actors usually have direct financial benefits from their involvement. This might also shape the SECO structure accordingly. The actor adaption model might be more closes than a FOSS because of market limitations or direct competition between actors. There is a stronger need for non technical actors to promote products, advice placing on the market(s), manage the technical processes and so on. All these can conclude in a bigger need for a hierarchical management model.

## 5 Conclusion

Software ecosystems is a field that has been gaining in popularity the last five years. The software industry is moving closer to the solution of software ecosystems with systems like the Google Android and Apple iOS increasing in popularity while the research world has increasing interest in the field with the third year of a dedicated workshop (IWSECO 2011). This report 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 until the current time and identify possible issues or areas not covered. In this literature review, we found and analyzed 59 relevant papers from a gross total of 212 extracted from a list of scientific libraries. We managed to provide an overview of the definition of SECO as it is mentioned in the bibliography, find patterns in the different definitions provided and list the common main items that consist a SECO. We reported an early increase in the research invested in the field 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 the SECO architecture by analyzing the three main components consisting the architecture, as

we deducted them from the literature. Finally, we examined the amount of research related to the industry by recognizing at the same time that a percentage of this amount was focusing on FOSS SECOS. Concluding we recognize that the field of SECO is still new in the research community and a lot of research is focusing on defining different perspectives and building the research knowledge-base for future work. Additionally, we recognized a lack of industrial studies in real and healthy SECOS.

## Acknowledgements

This work has been partially funded by the Net4Care project within Caretech Innovation<sup>9</sup>.

---

<sup>9</sup><http://www.caretechinnovation.dk/projekter/net4care/>

## References

- [1] Jan Bosch. From software product lines to software ecosystems. In *Proceedings of the 13th International Software Product Line Conference*, SPLC '09, pages 111–119, Pittsburgh, PA, USA, 2009. Carnegie Mellon University.
- [2] Joost te Molder, Ben van Lier, and Slinger Jansen. Clopenness of systems: The interwoven nature of ecosystems. In *Third International Workshop on Software Ecosystems (IWSECO-2011)*, pages 52–64. CEUR-WS, 2011.
- [3] D.G. Messerschmitt and C. Szyperski. Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books*, 1, 2005.
- [4] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. *Engineering*, 2(EBSE 2007-001), 2007.
- [5] M. Shaw. Writing good software engineering research papers. *Mini-tutorial for Proc ICSE' 03*, 2003.
- [6] Rick Kazman, Michael Gagliardi, and William Wood. Scaling up software architecture analysis. *Journal of Systems and Software*, In Press, Corrected Proof:–, 2011.
- [7] Daniel Hilkert, Christian M. Wolf, Alexander Benlian, and Thomas Hess. The as-a-service-paradigm and its implications for the software industry insights from a comparative case study in crm software ecosystems. In Will Aalst, John Mylopoulos, Norman M. Sadeh, Michael J. Shaw, Clemens Szyperski, Pasi Tyrvinen, Slinger Jansen, and Michael A. Cusumano, editors, *Software Business*, volume 51 of *Lecture Notes in Business Information Processing*, pages 125–137. Springer Berlin Heidelberg, 2010. 10.1007/978-3-642-13633-7\_11.
- [8] Ligu Yu, S. Ramaswamy, and J. Bush. Symbiosis and software evolvability. *IT Professional*, 10(4):56–62, july-aug. 2008.
- [9] Renato A. C. Capuruço and Luiz F. Capretz. Integrating recommender information in social ecosystems decisions. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 143–150, New York, NY, USA, 2010. ACM.
- [10] M. Lungu. Towards reverse engineering software ecosystems. In *Software Maintenance, 2008. ICSM 2008. IEEE International Conference on*, pages 428–431, 28 2008-oct. 4 2008.
- [11] O. Pettersson and D. Gil. On the issue of reusability and adaptability in m-learning systems. In *Wireless, Mobile and Ubiquitous Technologies in Education (WMUTE), 2010 6th IEEE International Conference on*, pages 161–165, april 2010.
- [12] Karl Michael Popp. Hybrid revenue models of software companies and their relationship to hybrid business models. In *Third International Workshop on Software Ecosystems (IWSECO-2011)*, pages 77–88. CEUR-WS, 2011.
- [13] L. Bass, P. Clements, and R. Kazman. *Software architecture in practice*. Addison-Wesley Longman Publishing Co., Inc., 2003.

- [14] Klaus Marius Hansen, Kristjan Jonasson, and Helmut Neukirchen. Controversy corner: An empirical study of software architectures' effect on product quality. *J. Syst. Softw.*, 84:1233–1243, July 2011.
- [15] Jan Bosch. Architecture challenges for software ecosystems. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 93–95, New York, NY, USA, 2010. ACM.
- [16] S. Jansen, A. Finkelstein, and S. Brinkkemper. A sense of community: A research agenda for software ecosystems. In *Software Engineering - Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages 187–190, may 2009.
- [17] Jan Bosch and Petra M. Bosch-Sijtsema. Softwares product lines, global development and ecosystems: Collaboration in software engineering. In Ivan Mistrk, Andr van der Hoek, John Grundy, and Jim Whitehead, editors, *Collaborative Software Engineering*, pages 77–92. Springer Berlin Heidelberg, 2010. 10.1007/978-3-642-10294-3\_4.
- [18] Jan Bosch and Petra Bosch-Sijtsema. From integration to composition: On the impact of software product lines, global development and ecosystems. *Journal of Systems and Software*, 83(1):67 – 76, 2010. SI: Top Scholars.
- [19] Mircea Lungu, Michele Lanza, Tudor Grba, and Romain Robbes. The small project observatory: Visualizing software ecosystems. *Science of Computer Programming*, 75(4):264 – 275, 2010. Experimental Software and Toolkits (EST 3): A special issue of the Workshop on Academic Software Development Tools and Techniques (WASDeTT 2008).
- [20] Jan Bosch and Petra Bosch-Sijtsema. Coordination between global agile teams: From process to architecture. In Darja Žmite, Nils Brede Moe, and Pr J. gerfalk, editors, *Agility Across Time and Space*, pages 217–233. Springer Berlin Heidelberg, 2010. 10.1007/978-3-642-12442-6\_15.
- [21] Jan Bosch. Architecture in the age of compositionality. In Muhammad Babar and Ian Gorton, editors, *Software Architecture*, volume 6285 of *Lecture Notes in Computer Science*, pages 1–4. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-15114-9\_1.
- [22] Rodrigo Pereira dos Santos and Claudia Maria Lima Werner. A proposal for software ecosystem engineering. In *Third International Workshop on Software Ecosystems (IWSECO-2011)*, pages 40–51. CEUR-WS, 2011.
- [23] Joey van Angeren, Jaap Kabbedijk, and Karl Michael Popp. A survey of associate models used within large software ecosystems. In *Third International Workshop on Software Ecosystems (IWSECO-2011)*, pages 27–39. CEUR-WS, 2011.
- [24] T.A. Alspaugh, H.U. Asuncion, and W. Scacchi. The role of software licenses in open architecture ecosystems. In *First International Workshop on Software Ecosystems (IWSECO-2009)*, pages 4–18. Citeseer, 2009.
- [25] Samuel Fricker. Specification and analysis of requirements negotiation strategy in software ecosystems. In *First International Workshop on Software Ecosystems (IWSECO-2009)*, pages 19–33. Citeseer, 2009.

- [26] R.S. Brewer and P.M. Johnson. Wattdepot: An open source software ecosystem for enterprise-scale energy data collection, storage, analysis, and visualization. In *Smart Grid Communications (Smart-GridComm), 2010 First IEEE International Conference on*, pages 91–95, oct. 2010.
- [27] T. Kakola. Standards initiatives for software product line engineering and management within the international organization for standardization. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10, jan. 2010.
- [28] H.C. An. Research on software problems based on ecological angle. In *Environmental Science and Information Application Technology, 2009. ESIAT 2009. International Conference on*, volume 3, pages 11–14, july 2009.
- [29] M. Lungu, J. Malnati, and M. Lanza. Visualizing gnome with the small project observatory. In *Mining Software Repositories, 2009. MSR '09. 6th IEEE International Working Conference on*, pages 103–106, may 2009.
- [30] Shuli Yu and C. Woodard. Innovation in the programmable web: Characterizing the mashup ecosystem. In George Feuerlicht and Winfried Lamersdorf, editors, *Service-Oriented Computing ICSOC 2008 Workshops*, volume 5472 of *Lecture Notes in Computer Science*, pages 136–147. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-01247-1\_13.
- [31] T. Janner, C. Schroth, and B. Schmid. Modelling service systems for collaborative innovation in the enterprise software industry - the st. gallen media reference model applied. In *Services Computing, 2008. SCC '08. IEEE International Conference on*, volume 2, pages 145–152, july 2008.
- [32] Walt Scacchi. Collaboration practices and affordances in free/open source software development. In Ivan Mistrk, Andr van der Hoek, John Grundy, and Jim Whitehead, editors, *Collaborative Software Engineering*, pages 307–327. Springer Berlin Heidelberg, 2010. 10.1007/978-3-642-10294-3\_15.
- [33] Walt Scacchi. Free/open source software development: Recent research results and methods. In Marvin V. Zelkowitz, editor, *Architectural Issues*, volume 69 of *Advances in Computers*, pages 243–295. Elsevier, 2007.
- [34] Walt Scacchi. Free/open source software development: recent research results and emerging opportunities. In *The 6th Joint Meeting on European software engineering conference and the ACM SIG-SOFT symposium on the foundations of software engineering: companion papers*, ESEC-FSE companion '07, pages 459–468, New York, NY, USA, 2007. ACM.
- [35] Mircea Lungu and Michele Lanza. The small project observatory: a tool for reverse engineering software ecosystems. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, ICSE '10*, pages 289–292, New York, NY, USA, 2010. ACM.
- [36] Rick Kazman and Hong-Mei Chen. The metropolis model and its implications for the engineering of software ecosystems. In *Proceedings of the FSE/SDP workshop on Future of software engineering*

- research, FoSER '10, pages 187–190, New York, NY, USA, 2010. ACM.
- [37] Samuel Fricker. Requirements value chains: Stakeholder management and requirements engineering in software ecosystems. In Roel Wieringa and Anne Persson, editors, *Requirements Engineering: Foundation for Software Quality*, volume 6182 of *Lecture Notes in Computer Science*, pages 60–66. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-14192-8\_7.
- [38] A. M. Alves and Marcelo Pessôa. Brazilian public software: beyond sharing. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '10, pages 73–80, New York, NY, USA, 2010. ACM.
- [39] Hendrik Brummermann, Markus Keunecke, and Klaus Schmid. Variability issues in the evolution of information system ecosystems. In *Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems*, VaMoS '11, pages 159–164, New York, NY, USA, 2011. ACM.
- [40] A. Hindle, I. Herraiz, E. Shihab, and Zhen Ming Jiang. Mining challenge 2010: FreeBSD, gnome desktop and debian/ubuntu. In *Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on*, pages 82–85, may 2010.
- [41] Krishna Raj P M and Srinivasa K G. Analysis of projects and volunteer participation in large scale free and open source software ecosystem. *SIGSOFT Softw. Eng. Notes*, 36:1–5, March 2011.
- [42] Walt Scacchi. The future of research in free/open source software development. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, FoSER '10, pages 315–320, New York, NY, USA, 2010. ACM.
- [43] Erik Yu and Stephanie Deng. Understanding software ecosystems: A strategic modeling approach. In *Third International Workshop on Software Ecosystems (IWSECO-2011)*, pages 65–76. CEUR-WS, 2011.
- [44] S. Jansen, S. Brinkkemper, and A. Finkelstein. Business network management as a survival strategy: A tale of two software ecosystems. In *First International Workshop on Software Ecosystems (IWSECO-2009)*, pages 34–48. Citeseer, 2009.
- [45] Ivo van den Berk, Slinger Jansen, and L{u}tzen Luinenburg. Software ecosystems: a software ecosystem strategy assessment model. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 127–134, New York, NY, USA, 2010. ACM.
- [46] Ivo Hunink, Rene van Erk, Slinger Jansen, and Sjaak Brinkkemper. Industry taxonomy engineering: the case of the european software ecosystem. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 111–118, New York, NY, USA, 2010. ACM.
- [47] Geir K. Hanssen. A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. *Journal of Systems and Software*, In Press, Accepted Manuscript:–, 2011.

- [48] Rodrigo Pereira dos Santos and Cláudia Maria Lima Werner. Revisiting the concept of components in software engineering from a software ecosystem perspective. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 135–142, New York, NY, USA, 2010. ACM.
- [49] P. R. J. Campbell and Faheem Ahmed. A three-dimensional view of software ecosystems. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 81–84, New York, NY, USA, 2010. ACM.
- [50] Vasilis Boucharas, Slinger Jansen, and Sjaak Brinkkemper. Formalizing software ecosystem modeling. In *Proceedings of the 1st international workshop on Open component ecosystems*, IWOCE '09, pages 41–50, New York, NY, USA, 2009. ACM.
- [51] Mohsen Anvaari and Slinger Jansen. Evaluating architectural openness in mobile software platforms. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 85–92, New York, NY, USA, 2010. ACM.
- [52] Olvano Barbosa and Carina ALaves. A systematic mapping study on software ecosystems. In *Third International Workshop on Software Ecosystems (IWSECO-2011)*, pages 15–26. CEUR-WS, 2011.
- [53] John D. McGregor. A method for analyzing software product line ecosystems. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 73–80, New York, NY, USA, 2010. ACM.
- [54] P. Schugerl, J. Rilling, R. Witte, and P. Charland. A quality perspective of software evolvability using semantic analysis. In *Semantic Computing, 2009. ICSC '09. IEEE International Conference on*, pages 420–427, sept. 2009.
- [55] Mathieu Goeminne and Tom Mens. A framework for analysing and visualising open source software ecosystems. In *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWSE)*, IWSE-EVOL '10, pages 42–47, New York, NY, USA, 2010. ACM.
- [56] Gerard Briscoe. Complex adaptive digital ecosystems. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '10, pages 39–46, New York, NY, USA, 2010. ACM.
- [57] Kurt Schneider, Sebastian Meyer, Maximilian Peters, Felix Schliephacke, Jonas Mrschbach, and Lukas Aguirre. Feedback in context: Supporting the evolution of it-ecosystems. In M. Ali Babar, Matias Vierimaa, and Markku Oivo, editors, *Product-Focused Software Process Improvement*, volume 6156 of *Lecture Notes in Computer Science*, pages 191–205. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-13792-1\_16.
- [58] Dominik Seichter, Deepak Dhungana, Andreas Pleuss, and Benedikt Hauptmann. Knowledge management in software ecosystems: software artefacts as first-class citizens. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 119–126, New York, NY, USA, 2010. ACM.
- [59] Oskar Pettersson, Martin Svensson, Didac Gil, Jesper Andersson, and Marcelo Milrad. On the role of software process modeling in

- software ecosystem design. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 103–110, New York, NY, USA, 2010. ACM.
- [60] Deepak Dhungana, Iris Groher, Elisabeth Schludermann, and Stefan Biffl. Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 96–102, New York, NY, USA, 2010. ACM.
- [61] Marcelo Cataldo and James D. Herbsleb. Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 65–72, New York, NY, USA, 2010. ACM.
- [62] Ligu Yu, S. Ramaswamy, and J. Bush. Software evolvability: An ecosystem point of view. In *Software Evolvability, 2007 Third International IEEE Workshop on*, pages 75–80, oct. 2007.
- [63] Tom Mens and Mathieu Goeminne. Analysing the evolution of social aspects of open source software ecosystems. In *Third International Workshop on Software Ecosystems (IWSECO-2011)*, pages 1–14. CEUR-WS, 2011.
- [64] Mircea Lungu, Romain Robbes, and Michele Lanza. Recovering inter-project dependencies in software ecosystems. In *Proceedings of the IEEE/ACM international conference on Automated software engineering*, ASE '10, pages 309–312, New York, NY, USA, 2010. ACM.