Software Ecosystems Governance:
A Systematic Literature Review and Research Agenda

Carina Alves[1], Joyce Oliveira[1] and Slinger Jansen[2]
[1]*Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil*
[2]*Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands*
{cfa, japo}@cin.ufpe.br, slinger.jansen@uu.nl

Keywords: Software Ecosystems, Governance, Health, and Systematic Literature Review.

Abstract: The field of Software ecosystems is a growing discipline that has been investigated from managerial, social, and technological perspectives. The governance of software ecosystems requires a careful balance of control and autonomy given to players. Orchestrators that are able to balance their own interests by bringing joint benefits for other players are likely to create healthy ecosystems. Selecting appropriate governance mechanisms is a key problem involved in the management of proprietary and open source ecosystems. This article summarizes current literature on software ecosystem governance by framing prevalent definitions, classifying governance mechanisms, and proposing a research agenda. We performed a systematic literature review of 63 primary studies. Several studies describe governance mechanisms, which were classified in three categories: value creation, coordination of players, and organizational openness and control. The number of studies indicates that the domain of software ecosystems and their governance is maturing. However, further studies are needed to address central challenges involved on the implementation of appropriate governance mechanisms that can nurture the health of ecosystems. We present a research agenda with several opportunities for researchers and practitioners to explore these issues.

1 INTRODUCTION

In the last decade, a large amount of research has been devoted to investigate the field of software ecosystems from managerial, social, and technological perspectives (Barbosa et al., 2013), (Bosch, 2014). Software ecosystems are sets of actors functioning as a unit and interacting with a shared market for software and services, together with relationships among them (Jansen et al., 2009).

A software ecosystem frequently relies on a platform on which extenders can build specific solutions to create complementary value (Jansen et al., 2012). Independent developers can extend and enrich the platform while sharing costs and risks with the platform owner. Examples of successful software platforms are Apple's iOS, Google Apps, and the Mozilla Firefox browser. The leading firm, typically called the orchestrator (or keystone) firm, must promote the sustainable development of the ecosystem by defining strategies and orchestrating the activities of players. The orchestrator is responsible for managing the evolution of the enterprise architecture (Iyer et al., 2007) and the interactions among all actors within the ecosystem (Manikas and Hansen, 2013a). The governance of software ecosystems requires a careful balance of control and autonomy given to players. Orchestrators that are able to balance their own interests by bringing joint benefits for other players are likely to create healthy ecosystems. Software ecosystems governance has become a crucial managerial aspect for proprietary platform owners and open source communities.

According to Tiwana (2010), governance mechanisms are employed to establish the level of control, decisions rights, and scope of proprietary versus shared ownership. There are several models to govern software ecosystems. For instance, GNU Linux is an open source ecosystem with a thriving community of developers. Apple's iOS is a prosperous example of proprietary ecosystem with tight control mechanisms. Google built a lively ecosystem around its Android open source community named the "Open handset Alliance". On the other hand, Nokia's Symbian is an open source operating system that failed to create a vibrant ecosystem due to its inability to attract partners and develop a rich set of apps (West and Wood, 2008). The examples above show that choosing the right ecosystem strategies and governance mechanisms are life-or-death decisions for orchestrator organizations. In fact, companies engaging in an ecosystem are mutually dependent on each other for survival (Yansiti and Levien, 2004).

We define software ecosystem governance mechanisms as managerial tools of participants in software ecosystems, i.e., orchestrators and platform extenders that have the goal of influencing an ecosystem's health. Ecosystems are healthy when they exhibit longevity and propensity for growth (Den Hartigh and Visscher, 2006).

Selecting appropriate governance mechanisms is not a trivial task. The challenge is to bound players actions without excessively constraining the desired level of innovation and value creation in the ecosystem. This situation creates fine tension between control and autonomy. Balancing these tensions is one of the main goals of software ecosystem governance. The correct implementation of governance mechanisms can accommodate these tensions towards a sustainable and healthy ecosystem. On the other hand, ineffective governance can result in a declining growth of the ecosystem (Wareham et al., 2014). The challenge of selecting ecosystem governance strategies that contributes towards the ecosystem health has driven us to conduct a systematic literature review. Our review aims at synthetizing the increasing number of studies in the field of software ecosystem governance.

This article is organized as follows. Section 2 describes the research method. The results of the review are presented in Section 3. To discuss the results of our review and propose future areas for investigation, a research agenda containing six areas of interest is proposed in Section 4. Then, we discuss threats to validity in Section 5. Finally, Section 6 concludes this article.

2 RESEARCH METHOD

A Systematic Literature Review (SLR) is a means for answering specific research questions, examining a particular research topic, or phenomenon of interest by systematically identifying, evaluating, and interpreting available relevant research. Our review protocol follows guidelines from Kitchenham and Charters (2007). We undertook the review of studies following these activities: defining research questions, searching relevant studies, applying inclusion/exclusion criteria, assessing the quality of studies, analysing data, and synthesis.

2.1 Research Questions

We specified two research questions to guide our study:

- RQ1. How is governance characterized in software ecosystems literature?
- RQ2. What are the mechanisms proposed to govern software ecosystems?

Governance is a well-established concept primarily associated with the needs to protect investment and ensure the sustainability of businesses through time (Hoogervorst, 2009). Corporate governance refers to the mechanisms, processes, and relations by which corporations are controlled and governed (OECD, 2004). Governance involves a set of principles to direct the distribution of rights and responsibilities among stakeholders. In RQ1, we present and discuss available definitions for software ecosystems governance proposed by primary studies. Then, we compare the definitions available and propose an integrated definition for the term.

Traditional corporate governance mechanisms include monitoring actions, policies, and decisions by aligning the interests of different stakeholders. According to Croteau et al. (2013), IT governance can be organized by the attributes: structure, process, and participants. In this SLR, we opted not to follow a pre-existing classification. Instead, we classify the governance mechanisms based on the data gathered from the primary studies following a thematic analysis approach (Cruzes and Dybå, 2011). Our goal to answer the second question (RQ2) is to identify the mechanisms proposed by current literature to govern software ecosystems.

2.2 Search Process

To guide the systematic literature review, a protocol was developed to specify the steps and criteria to undertake the review. The review protocol includes details of how different types of studies will be located, appraised, and synthesized (Brereton et al., 2007).

The strategy to collect studies included the following steps: (i) automatic search of electronic databases (ii) manual search of journals, conferences, and workshops (iii) analysis of reference lists from other secondary studies in software ecosystems. The automatic search was executed on the following databases: ACM Digital

Library, IEEE Xplore Digital Library, Science Direct, and SpringerLink. We used two independent search strings: "software ecosystem", "platform ecosystem". We opted to use generic terms to avoid over restricting the search process. In the early stages of our research we tried to use the search string "software ecosystem" AND "governance". However, we considered that using these combined keywords the results retrieved from the search engines were very limited. In addition, we conducted a manual search in the following journals, conferences, and workshops:

- Information and Software Technology;
- Journal of Software Systems;
- International Conference on Software Business;
- International Workshop on Software Ecosystems.

To complement our manual search, we analysed the references of the following reviews in the field of software ecosystems: (Barbosa et al., 2013; Franco-Bedoya et al, 2014; Manikas and Hansen, 2013a; Manikas, 2016). Although the scope and research questions of these reviews are different from ours, we examined the list of articles to correct any eventual omission of studies from the other search procedures. 7 studies [S13, S17, S18, S25, S26, S41, S61] were obtained from the analysis of secondary studies described above. Finally, we collected additional three studies [S31, S53, S57] that were recommended by experts in the field.

2.3 Inclusion and Exclusion criteria

We adopted the following inclusion criteria to select articles: (i) studies written in English, (ii) studies that answer at least one research question. The exclusion criteria adopted was: (i) secondary studies (e.g. mapping studies and systematic literature reviews), (ii) technical reports, abstracts, and whitepapers, (iii) duplicate reports of the same study.

The literature collection started with 997 articles returned from the electronic and manual search. The automatic search was conducted on the 5th of January 2016. We did not restrict year range in our search. Then, we excluded articles based on titles and abstract that did not satisfy our inclusion criteria. In practice, we assessed if the title and abstract are likely to answer at least one RQ. Whenever we were in doubt we included the article for further analysis of its full content. After this step, we included 592 studies. Then we read the full content of the articles and selected 67 primary studies. In a final step, a quality assessment of each article was conducted and we finally selected 63 articles (studies are listed in the Appendix).

The quality assessment criteria based on Brhel et al., (2015) are described below:

- Is there a clear statement of the research goals, e.g. in an explicitly verbalized research question?
- Is there an adequate description of the context in which the research was carried out?
- Only applicable to empirical research articles:
  - Is the research method explicitly stated?
  - Which research method was chosen?

2.4 Data extraction and analysis

We used a database to store data from the selected studies. Two researchers extracted data from the studies. Several discussion meetings were held with all authors to compare extractions, clarify uncertainties, agree on discrepancies, and perform sanity checks. To answer RQ1, we simply searched the term "governance" in the primary studies and checked if the article provided a definition for governance in the context of software ecosystems. To answer RQ2, we used thematic analysis as synthesis method, following the recommended steps proposed by (Cruzes and Dybå, 2011). We identified the relevant codes from primary studies. Then, we merged the codes into key themes. We considered that governance mechanisms were encapsulated in related terms, such as: "manage", "govern", "control", "strategy", "orchestration" and "critical factors".

3 RESULTS

3.1 Overview of Studies

63 studies were identified in our systematic literature review, as listed in the Appendix. Our final list included articles published between 2002 and 2016. We observed an increasing number of studies published over the last few years, where the peak publication period is from 2012 to 2015 (36 articles, 57%).

The most popular publication channels are: ICSOB (8 articles, 13%), IWSECO (5 articles, 8%), ECSA (3 articles, 5%), IST (5 articles, 8%), JSS (2 articles, 3%). In particular, the following events are dedicated to the field of ecosystems: IWSECO, MEDES, EWSECO, WEA, DEST. These results confirm that the field of software ecosystems has been receiving a growing attention from the academic communities of information systems, software business, and software engineering.

Figure 1 presents the main software ecosystems investigated by the primary studies. Apple and Android are the most frequent ecosystems examined (with 4 studies both ecosystems), followed by Siemens, IBM, GX Software and Eclipse (3 studies each ecosystem).

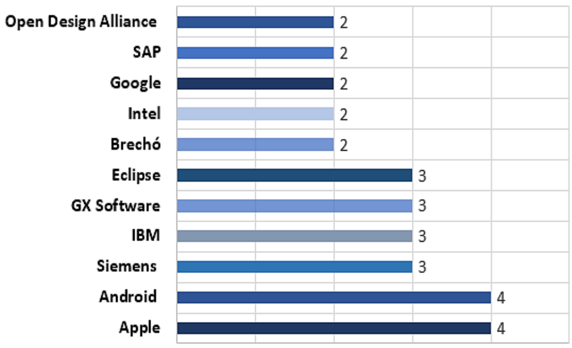

Figure 1. Software ecosystems investigated by the studies

We noted that both open source and proprietary ecosystems have been equally analysed. It is a good sign that researchers are focusing on the specific managerial issues faced by open source and proprietary ecosystems. The diversity of ecosystems being investigated reinforces the wide perspective that the software ecosystems community is gaining regarding the differences between open source and proprietary ecosystems.

Figure 2 shows the research type followed by the studies. We adopted the classification proposed by Wieringa et al., (2006). Petersen et al., (2015) provides helpful decision criteria on how to classify the studies. According to the classification, studies can be classified into six research types, namely:

Solution Proposal – The article proposes a novel solution or a significant improvement of an existing technique without a full validation.

Philosophical Paper – The paper proposes a conceptual framework and a new way to look at things.

Opinion Papers – The paper presents the author's opinion about something.

Experience Paper – The paper describes the practical experience of the author who is generally an industry practitioner.

Validation Research – The paper describes an empirical validation of a solution done in the lab. Examples of studies include: student case study, mathematical analysis, prototyping, laboratory experiments, and simulation.

Evaluation Research – The paper presents a real-world industrial evaluation of a solution. It includes: industrial case study, controlled experiments with practitioners, practitioner targeted survey, action research, and ethnography.

Figure 2. Research design adopted by the studies

The most common type of study identified in our review is evaluation research (27 studies). This result suggests that the majority of studies explore the field from a practical perspective by conducting empirical studies of real ecosystems. 21 studies are classified as philosophical papers. 11 studies propose a solution such as new techniques, models, and methods. 2 studies present validation research. Finally, we identified 2 experience papers. We did not include any opinion paper in our list of primary studies.

3.2 Answering RQ1: How is governance characterized in software ecosystems literature?

Our results show that the concept of governance is gaining importance in software ecosystem literature. 9 studies [S1, S4, S20, S21, S28, S32, S51, S54, S62] explicitly define what is software ecosystem governance. Jansen [S34] proposes that governance is one of the key domains of the Open Software Enterprise Model. The study adopts the definition of governance given by Dubinsky and Kruchten [S12], who consider governance as "*the way an* *organization is managed, including its powers, responsibilities, and decision-making processes*". According to Jansen et al. [S32], it also involves the assignment of roles and decision rights, measures, and policies. A fundamental governance decision that orchestrators must make is how much power is given to the community and how much control it keeps for itself. Jansen and Cusmano [28] and van Angeren et al. [S54] consider that ecosystem governance "*involves the use of strategic procedures and processes to control, maintain, or change the ecosystem*". Study [S54] also states that software ecosystems governance "*encompasses both technical and managerial aspects, including the management of the software platform and its interfaces, definition of business and partnership models, and establishment of entry barriers*".

Baars and Jansen [S4] propose a framework for software ecosystems governance. They define software ecosystems governance as: "*procedures and processes by which a company controls, changes or maintains its current and future position in a software ecosystem on all different scope levels*". Studies [S1], [28] and [S62] adopt the same definition. Albert and colleagues [S1] present a software ecosystem governance approach for enabling IT architecture based on software asset management. In [S28], Jansen and Cusumano propose a governance model for ecosystem health preservation and improvement. Wnuk et al., [S62] evaluate the model proposed in [S28] by means of a case study in a hardware-dependent software ecosystem.

We noted that several primary studies discuss the classical tension between open and closed governance models. Jansen et al., [S32] suggest that companies benefit from opening-up their business models. It includes sharing strategic knowledge, making the ecosystem strategy explicit to all players, and coordinating actions. The authors propose an openness degree to assess how open a company is. In [S13], the authors do not define what is governance in the context of ecosystems, but they provide a rich discussion on the tensions between open and closed governance models as platforms mature. The study proposes that hybrid governance models are more suitable for both proprietary and shared platforms. Such model is characterized by the centralized control over platform technology and shared responsibilities for the ecosystem community.

Ghazawne et al., [S20] argue that the governance of platform ecosystems involves "*a delicate balance act of the platform owner, trying to keep control of the platform while simultaneously seeking to expand the diversity of potential developers*". According to Tiwana et al. [S51], governance broadly refers to "*who decides what in an ecosystem*". Study [S51] investigates the evolution of platform-centric ecosystems and proposes that governance can be analyzed from three facets: i) how decision rights are divided between the platform owner and app developers, ii) what types of formal and informal control mechanisms are used by the platform owner, and iii) how ownership is regulated if the platform is property of a single company or shared by multiple owners. [S51] also states that ecosystem governance "*involves sharing responsibilities and authority, aligning incentives, and sharing stakes*". Goldbach and Kemper [S21] adopt the same definition of platform governance given by study [S51] to understand how control mechanisms imposed by the platform owner affects the platform stickiness. All primary studies that answer this research question suggest that a key challenge faced by platform owners is balancing their own strategic objectives with the goals and activities of players within the platform. Such delicate balance becomes critical for software ecosystems to thrive.

7 studies [S10, S28, S32, S41, S54, S55, S62] indicate that ecosystem governance influences the health and sustainability of ecosystems. This means that governance strategies and managerial decisions taken by orchestrators will affect the healthy evolution of the entire ecosystem. The primary studies suggest that health metrics provide operational indicators on how software ecosystems are governed. For instance, if an open governance model is adopted by the ecosystem, more autonomy will be given to players to shape their future growth and expansion. Otherwise, in a closed governance model, the orchestrator holds substantial power and control over the players. Consequently, the orchestrator has more responsibility towards the prosperity and overall health of the ecosystem. Defining the openness strategy is an important decision that orchestrators must make when structuring the governance model for their software ecosystems. This decision will have a significant impact on the evolution of the enterprise architecture of integrated systems. In particular, players must decide if their enterprise architecture will follow a centralized, federated or decentralized organizational structure (Rychkova et al., 2013). We conclude this section by proposing an integrated definition for software ecosystems governance: *all processes by which a player creates value, coordinates relationships, and defines controls.*

3.3 RQ2: What are the mechanisms proposed to govern software ecosystems?

We define software ecosystem governance mechanisms as managerial tools of players in software ecosystems that have the goal of influencing an ecosystem's health. We observed that frequently authors use terms such as “orchestration” and “management” to refer to what can be understood as a governance mechanism. To classify the 63 studies, we propose three main categories of governance mechanisms:

Value Creation – involve mechanisms to generate and distribute value for the whole ecosystem. Value creation mechanisms are generally proposed and nurtured by the orchestrator (i.e. platform and/or marketplace owner), who must understand how to create value that is appreciated both by partners and customers. In this context, it is important to identify sources of value (such as licenses and revenue models), and stimulate the co-creation of value among players, by means of innovation, investments, and cost sharing. As a result, the ecosystem can attract and retain partners who will mutually benefit from the value distributed within the ecosystem. This category covers all the incentives and benefits that players can gain from a software ecosystem.

Coordination of Players – describe mechanisms to maintain the consistency and integration of activities, relationships, and structures of the ecosystem, for both customers and partners, leading to a harmonious and effective coordination with players in the ecosystem. We identified mechanisms to stimulate partnership models, define roles and responsibilities for players, improve communication channels within the ecosystem, and nurture collaborations. In addition, primary studies propose mechanisms to manage critical issues, such as: conflicts, resources, risks, and expectations. This category focuses on the coordination aspects of governance, whereas the next focuses on strategic decisions of openness and control.

Organizational Openness and Control – these mechanisms capture the notorious tension between open versus closed organizational models and represent how control will be retained by the orchestrator to guarantee its power position and how autonomy will be given for the community to make their own decisions independently. On the one side, orchestrators can support autonomy, distribute power, and share knowledge regarding technological roadmaps and architectural decisions. On the other side, orchestrators can keep control by defining entry requirements, establishing quality standards, and through certifications.

Table 1 shows the classification of governance mechanisms proposed by the primary studies. We observed that the most cited mechanisms are: attract and maintain partners (28 articles, 44%), share knowledge (20 articles, 31%), promote innovation (25 articles, 39%), manage licenses (21 articles, 33%). We do not claim that these are the most important governance mechanisms, as several studies suggest that the governance must match the specific context and market drivers involved in the ecosystem [S4, S13, S25 S32, S43].

4 DISCUSSION AND RESEARCH AGENDA

The synthesis provided by our literature review enables further analysis and insights regarding the future role of software ecosystems governance in software producing organizations. We express the following needs, which are requirements that should to be met to advance the field of software ecosystems governance. The following statements can contribute to the overall research agenda on software ecosystems and serve as an addendum to the works of Jansen et al (2009), Barbosa et al (2013), Manikas (2016), and Axelsson and Skoglund (2016).

1. The Need for a Common Vocabulary in Software Ecosystems Governance - The numbers of publications in this domain emphasize that the field of software ecosystems governance is maturing. Increasing numbers of work take positions on definitions of the concepts central to software ecosystems: health (Manikas and Hansen, 2013b), governance (this work), open source ecosystems [S34], developer ecosystems [S33], and quality in software ecosystems (Axelsson and Skoglund, 2016), each of these concepts is settling in as an established term in the ecosystems discourse. We identified that several studies adopt related terms such as management and orchestration to refer to governance mechanisms. Therefore, we suggest the need to establish a common glossary and conceptual framework that collects these definitions into one tome of ecosystems governance knowledge.

Table 1: Governance Mechanisms in Software Ecosystems

Governance Mechanisms		Studies	Number of Studies
Value Creation	Promote innovation	S61, S7, S32, S50, S40, S48, S52, S9, S47, S3, S7, S45, S10, S8, S35, S27, S18, S17, S19, S61, S38, S57, S24, S43, S44	25 (39%)
	Manage licenses	S16, S32, S41, S40, S1, S3, S6, S58, S28, S2, S51, S63, S8, S27, S18, S13, S17, S57, S31, S22, S24	21(33%)
	Create revenue models	S7, S3, S45, S58, S4, S5, S7S6, S28, S10, S62, S30, S23, S27, S61, S38, S53, S57, S23, S36, S39	20 (31%)
	Attract and maintain varied partners	S61, S32, S29, S52, S47, S45, S15, S58, S4, S6, S10, S62, S55, S46, S63, S35, S27, S18, S17, S61, S38, S53, S57, S42, S19, S23, S26, S36	28(44%)
	Stimulate partner investments and share costs	S61, S56, S3, S45, S8, S27, S22, S23, S43	9 (14%)
Coordination of Players	Create partnership models	S32, S56, S54, S4, 28, S62, S55, S49, S30, S8, S27, S53, S31, S19, S24	15 (23%)
	Define rules to manage relationships	S32, S40, S29, S56, S52, S9, S3, S4, S5, S46, S2, S63, S35, S27, S57, S42, S36	17 (26%)
	Establish roles and responsibilities	S41, S50, S40, S56, S3, S15, S4, S5, S49, S46, S51, S63, S27, S13, S42, S26, S37	17 (21%)
	Enable effective communication channels	S41, S29, S48, S52, S9, S3, S11, S14, 28, S16, S27, S31, S37	13 (20%)
	Manage conflicts	S32, S52, S15, S8, S27, S57, S31, S42, S19	9 (14%)
	Manage resources	S1, S52, S9, S47, S3, S15, S10, S46, S20, S35, S42, S26, S36, S44	14 (22%)
	Manage risks	S50, S40, S56, S52, S58, S46 , S30, S8, S18, S17, S57, S22, S39, S43	14 (22%)
	Manage expectations	S47, S49, S16	3 (4%)
	Nurture collaborations	S61, S50, S52, S46, S58, 28, S62, S55, S49, S46, S35, S17, S42, S44	14(22%)
Organizational Openness and Control	Support autonomy	S7, S50, S52, S3, S48, S4, S7, S46, S20, S51, S35, S18, S17, S61, S42	15(23%)
	Share knowledge	S16, S32, S50, S40, S29, S48, S52, S3, S4, S11, S62, S30, , S20, S35, S18, S17, S61, S57, S31, S37	20 (31%)
	Distribute power	S32, S50, S52, S3, S15, S46, S16, S51, S27, S37	10 (15%)
	Define entry requirements	S54, S45, S4, S28, S62, S30, S18, S38, S53, S24, S36	11 (17%)
	Share architectural decisions	S16, S29, S48, S1, S52, S9, S47, S3, S5, S58, S28, S62, S2, S51, S27, S11, S14	17 (21%)
	Share roadmaps	S52, S58, S28, S27, S57, S31	6 (9%)
	Define quality standards and certifications	S32, S41, S50, S40, S56, S58, S28, S62, S55, S30, S38, S57, S22	13 (20%)

2. The Need for Practical Governance Guidance - Even though there exists an extensive body of knowledge on software ecosystems governance, it is hard for practitioners to extract practical and strategic guidance from the works under study. There is a need for more consumable and practical knowledge for practitioners. Other relevant studies for practitioners interested on creating health ecosystems dashboards, include Goeminne and Mens (2013) on GitHub analysis, collecting intelligence on the progress of particular ecosystems. These tools can form the basic groundwork under mature evaluation mechanisms and tools for large open and commercial software ecosystems.

3. The Need for Analysing the Interplay between Governance Mechanisms and Health Metrics – Our study indicates that health metrics provide operational indicators on how software ecosystems are governed. Therefore, by selecting appropriate health metrics, players can govern the ecosystem towards a sustainable path. A challenge remains on how to implement governance to foster innovation and encourage autonomous behaviour for diversity, without undermining the quality of software and accountability of players' actions [S20]. The tension between control and autonomy must be appropriately balanced. Understanding how the implementation of specific governance mechanisms affects the success of ecosystems and the underlying enterprise platform is an exciting problem for scholars in the field.

4. The Need for Understanding the Governance of Developer Ecosystems - The developers' and niche players' impacts in ecosystems are amplified by the success of the ecosystem. Examples like Farmville for Facebook and Angry Birds for iOS illustrate how ecosystems grow immensely through the success of its constituents. The developers are the starting point for any software ecosystem; hence the recent increase of interest in developer ecosystems. There is a need for further understanding developers interests and behaviours [S38]. Barriers to entry, platform stickiness, and developer attraction are factors that require further research. An extension to this perspective is a need for further study of enterprise architecture and delivery mechanisms that enable software ecosystems [S33]. Orchestrators must understand developers' motivations and expectations to adopt appropriate governance mechanisms.

5. The Need for Studying Governance in Open Software Ecosystems - Open source ecosystems exhibit different properties than more traditional closed and commercial ecosystems. The openness of a platform permeates through every aspect of an ecosystem, whether it is about ownership of the code or about mechanisms around supporting tools, such as application stores. These openness questions also play a part in the architecture of a platform itself: without an open platform architecture, extenders cannot extend it. In our SLR, we found no study that presented a comparative analysis of governance mechanisms employed by open source versus proprietary ecosystems. This is a promising line of research.

6. The Need for Understanding the Interactions Between Ecosystems – Even good governance can lead to the demise of an ecosystem due to external factors. When looking at the governance and health of the Symbian ecosystem in 2007, it would have been hard to predict its demise. One can speculate about its poor business support from Nokia and fundamental faults in the business model of Symbian. However, it is hard to ignore the impending doom coming from the iPhone after 2007: its high rate of adoption and superior technology simply blew the Symbian ecosystem away. The challenge for governance research in the next decade will be to analyse and understand the interplay between large ecosystems. As long as standards, age-old ecosystems, and settled industry stacks can be blown away or grow exponentially through the workings of other ecosystems, we must develop governance tools and management practices that focus on the robustness of software ecosystems that can prepare for surviving in such storms.

5 THREATS TO VALIDITY

Our study faced similar validity threats as any other systematic literature review. Two of the main limitations in a review are the bias in selection and data extraction procedures (Kitchenham and Charters, 2007). Software ecosystem is a multidisciplinary field covering studies from software engineering, information systems, organization, and management science. To limit the threat of not including relevant primary studies, we adopted a search strategy with generic keywords to retrieve as many articles as possible that were related to the research topic. We complemented the automatic search with manual searches in the main journals, conferences, and workshops where studies in software ecosystems have been published. In addition, we also analysed the primary studies of other literature reviews published in the field.

In order to mitigate the impact of selection bias, we defined the review protocol with clear inclusion and exclusion criteria for each selection step. In the first selection step, a large number of irrelevant studies were removed by analysing title and abstract. One author performed this task. In the second selection step, two authors screened the content of studies and constantly crosschecked the preliminary selection results. We also analysed the potential primary studies against a quality assessment checklist. With respect to bias in the data extraction, we had some problems to extract relevant information from primary studies. This problem was more critical to answer RQ2. We observed that studies use different terminologies to describe aspects related to governance mechanisms and metrics to operationalize health. This specific limitation of the software ecosystem literature was discussed on item 1 of our research agenda presented in Section 4. In several occasions we had to interpret the subjective information provided by the articles. To minimize interpretation bias, we conducted a very careful reading and had several discussion meetings among the authors during the data extraction phase.

6 CONCLUSION

The governance of software ecosystems is currently one of the largest challenges software platform companies need to deal with for the sake of their survival. Governance includes technical decisions regarding the enterprise architecture, social aspects involving the coordination of players, and business strategies. Therefore, governance impacts all three dimensions of software ecosystems (i.e. technical, social and business).

From 63 studies analysed, we conclude that software ecosystems governance is defined as all processes by which a player creates value, coordinates relationships, and defines controls. An overview of software ecosystem governance mechanisms is provided, in which we classify governance mechanisms as belonging to mechanisms for value creation, coordination of players, and organizational openness and control. We identify approximately 20 governance mechanisms that can be directly implemented by players in software ecosystems and studied by the research community. To our knowledge there is no study in the field to systematically structure and classify governance mechanisms reported in the literature.

The practical impact of this work is that for practitioners in the software industry light is shed on the concept of ecosystem governance. Although we do not claim that the overview of governance mechanisms is complete, it provides a useful strategic tool for practitioners and a conceptual base for researchers. The scientific impact of the work is threefold: we provide insight into the concepts of software ecosystems governance and present mechanisms to perform governance. In future research, we aim at developing a governance conceptual model for software ecosystems.

REFERENCES

Axelsson, J., and Skoglund, M., 2016. Quality assurance in software ecosystems: A systematic literature mapping and research agenda. Journal of Systems and Software, Volume 114, pp. 69-81. 2016.

Barbosa, O., R.; Santos, R., C.; Alves, C., 2013; Werner, S.; Jansen. A Systematic Mapping Study on Software Ecosystems through a Three-dimensional Perspective. In Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. UK, Edward Elgar Publishing, pp. 59-81.

Bosch. J., and Bosch-Sijtsema, P.M. 2014. Esao: A ‚Holistic Ecosystem-Driven Analysis Model. In Proceedings of the 5th International Conference on Software Business. 2014.

Brereton, P., Kitchenham, B., Budgen, D., Turner, M., Khalil, M., 2007. Lesson from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software, Vol 80.

Brhel, M., Meth, H., and Maedcher, A., 2015. Exploring principles of user-centered agile software development: A literature review. Information and Software Technology, Vol 61.

Cobit 4.1. ISACA - Portuguease. 2007. ISACA / IT Governance Institute.

Croteau, A. Bergeron, F. Dubsky, J. 2013. Contractual and Consensual Profiles for an Interorganizational Governance of Information Technology. International Business Research. Vol. 6 n. 9.

Cruzes D., and Dybå, T., 2011. Recommended steps for thematic synthesis in software engineering. In Proceedings of International Symposium on Empirical Software Engineering and Measurement.

Den Hartigh, M., and Visscher, W., 2006. The health measurement of a business ecosystem. In proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting. pp. 1-39.

Franco-Bedoya, O., Ameller, D., Costal, D., and Franch. Queso., X. 2014 A quality model for open source software ecosystems. In Proceedings of the 9th International Conference on Software Engineering and Applications (ICSOFT-EA'14), pp. 209-221.

Goeminne, M., Claes, M, T., Mens, A., 2013. A historical dataset for the Gnome ecosystem. In Proceedings of the 10th IEEE Working Conference on Mining Software Repositories. 2013.

Gueguen, G., and Pellegtin-Boucher, E., 2004 Coopetition and IT business ecosystems: the success of SAP. the European Institute for Advanced Studies in Management.

Hoogervorst, J.A.P, 2009. Corporate Governance. In: Enterprise Governance and Enterprise Engineering. Springer. Volume 3, Issue 1, pp.3-11.

Iansiti, M., and Levien, R., 2004. Strategy as ecology. Harvard business review. Vol 82, Issue 3.

Iyer, B., Dreyfus, D., and Gyllstrom, P., 2007. A Network-based View of Enterprise Architecture. Handbook of Enterprise Systems Architecture in Practice, pp. 500-525.

Jalali, S., and Wohlin, C., 2012. Systematic literature studies: database searches vs. backward snowballing.

In International symposium on Empirical software engineering and measurement.

Jansen, S., Finkelstein, A., and Brinkkemper, S., 2009. A sense of community: A research agenda for software ecosystems. In Proceedings of the 31st Conference on Software Engineering, pp.187-190.

Kitchenham, B., and Charters, S., 2007. Guidelines for Performing Systematic Literature Review in Software Engineering. EBSE Technical Report, 2.3, Keele University.

Manikas, K., and Hansen, K.M., 2013a. Software Ecosystems – A Systematic Literature Review. Journal of Systems and Software, Volume 86, Issue 5, pp. 1294-1306.

Manikas, K., 2016. Revisiting Software Ecosystems Research: A Longitudinal Literature Study. Journal of Systems and Software. Vol 117.

Manikas, K., and Hansen, M.K., 2013b. Reviewing the Health of Software Ecosystems-A Conceptual Framework Proposal. In proceedings of the 5th International Workshop on Software Ecosystems (IWSECO'13), in ICSOB, pp. 33-44.

OECD. 2004. OECD Principles of Corporate Governance. OECD - Organisation for Economic Co-operation and Development. http://www.oecd.org/daf/ca/oecdprinciplesofcorporategovernance.htm.

Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for Conduction Systematic Mapping Studies in Software Engineering: An Update. Information and Software Technology. Vol 56/11. 2015.

Rychkova, I., Zdravkovic, J., & Speckert, T. 2013. Challenges of EA Methodologies Facing Progressive Decentralization in Modern Organizations. In PoEM (Short Papers) (pp. 18-28).

Tiwana, A., Konsynski, B., and Bush, A.A., 2010. Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. Information Systems Research. Volume 21, Issue 4, pp. 675–687. 2010.

Wareham, J., Fox, P.B., and Cano Giner, J.L, 2014. Technology ecosystem governance. Organization Science Volume 25, Issue 4, pp 1195-1215.

West, J., and Wood, D., 2008. Creating and Evolving an Open Innovation Ecosystem: Lessons from Symbian Ltd. Available at SSRN Working Paper Series.

Wieringa, R., Maiden, N., and Mead, N., 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Engineering Journal, Vol 11, Issue 1.

APPENDIX

S1. Albert. B.E., Santos, R.P., Werner, C., 2013. Software ecosystems governance to enable IT architecture based on software asset management. In Proceedings of the 7th IEEE International Conference on Digital Ecosystems and Technologies.

S2. Alspaugh, T.A., Asuncion, H. U., Scacchi, W., 2010. The Role of Software Licenses in Open Architecture Ecosystems. In proceedings of the 1st International Workshop on Software Ecosystems (IWSECO'09), pp. 4-18.

S3. Axelsson, J., Papatheocharous, E., Andersson, J., 2014. Characteristics of software ecosystems for Federated Embedded Systems: A case study. Information and Software Technology, Volume 56, Issue 11, pp.1457-1475.

S4. Baars, A., Jansen, S., 2012 A framework for software ecosystem governance. In Proceedings of the Third International Conference on Software Business (ICSOB'12), pp.168-180.

S5. Bosch, J., Bosch-Sijtsema, M.P., 2014. Esao: A Holistic Ecosystem-Driven Analysis Model. In Proceedings of the 5 th International Conference on Software Business (ICSOB 2014), Springer Verlag, pp. 179-193.

S6. Burkard, C., Widjaja, T., Buxmann, P., 2012. Software Ecosystems. Business & Information Systems Engineering, volume 4, Issue. 1, pp. 41-44.

S7. Campbell, P., Ahmed, F., 2010. A three-dimensional view of software ecosystems. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (ECSA '10), Carlos E. Cuesta (Ed.). ACM, New York, NY, USA, pp.81-84.

S8. Ceccagnoli, M., Forman, C., Huang, P., Wu, D.J., 2011. Co-creation of value in a platform ecosystem: The case of enterprise software. MIS Quarterly, Forthcoming.

S9. Che, M., Perry, D.E., 2014. Architectural Design Decisions in Open Software Development: A Transition to Software Ecosystems. In Proceedings of the 23rd Australian Software Engineering Conference, pp.58-61.

S10. Den Hartigh, E., Tol, M., Visscher, W., 2006. The health measurement of a business ecosystem. In proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting. pp. 1-39.

S11. Dittrich, Y., 2014. Software engineering beyond the project – Sustaining software ecosystems. Information and Software Technology, Volume 56, Issue 11, pp. 1436-1456.

S12. Dubinsky, Y., Kruchten, P., 2009. Software development governance (sdg): report on 2nd workshop. ACM SIGSOFT Software Engineering Notes, Volume 34, pp. 46-47.

S13. Eisenmann, T.R., Parker, G., M.W.; Van Alstyne,M.W., 2008. Opening platforms: how, when and why? In Platforms, Markets and Innovation. Gawer A (ed). Edward Elgar: Cheltenham, UK, pp. 131-162.

S14. Eklund, U., Bosch, J., 2012. Introducing software ecosystems for mass-produced embedded systems.

In Software Business. Springer Berlin Heidelberg, pp. 248-254.
S15. Fricker. S., 2010. Requirements Value Chains: Stakeholder Management and Requirements Engineering in Software Ecosystems. In proceedings of Requirements Engineering: Foundation for Software Quality (REFSQ'10), volume 6182, pp. 60-66.
S16. Fricker, S., 2009. Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems. In proceedings of Workshop on Software Ecosystems, pp. 19-33.
S17. Gawer, A., Cusumano, M.A., 2014. Industry platforms and ecosystem innovation. Journal of Product Innovation Management, Volume 31, Issue 3, pp. 417-433.
S18. Gawer, A., Henderson, R., 2007. Platform owner entry and innovation in complementary markets: Evidence from Intel. Journal of Economics & Management Strategy, Volume 16, Issue 1, pp. 1-34.
S19. Gawer, A., Cusumano, M.A., 2002. Platform leadership: How Intel, Microsoft, and Cisco drive industry innovation. Harvard Business School Press, pp. 29-30.
S20. Ghazawne, A., Henfridsson, O., 2010. Governing third-party development through platform boundary resources. In proceedings of International Conference on Information Systems (ICIS' 10), AIS Electronic Library (AISeL), pp 1-18.
S21. Goldbach, T., Kemper, V.K., 2014. Should I Stay or should I go? the effects of control mechanisms on app developers´ intention to stick with a platform. In proceedings of European Conference on Information Systems (ECIS'14).
S22. Haile, N., Altmann, J., 2016. Value creation in software service platforms. Future Generation Computer Systems, Volume 55, pp. 495-509.
S23. Hyrynsalmi, S., Suominen, A., Makila, T.,Jarvi, A., Knuutila, T., 2012. Revenue models of application developers in android market ecosystem. In M. Cusumano, B. Iyer, N. Venkatraman, eds.: ICSOB 2012. Number 114 in Lecture Notes in Business Information Processing. Springer Heidelberg.
S24. Huang, P., Ceccagnoli, M., Forman, C., Wu, D.J., 2013. Appropriability mechanisms and the platform partnership decision: Evidence from enterprise software. Management Science, Vol 59, Issue 1..
S25. Iansiti, M., Levien, R., 2004. Strategy as ecology. Harvard business review. Vol 82, Issue 3.
S26. Iyer, B., Lee,C.H., Venkatraman, N., 2006.. Managing a small world ecosystem: Some lessons from the software sector. California Management Review, Volume 48, Issue 3, pp. 28-47.
S27. James, W.P., and De Meyer, A., 2012. Ecosystem advantage: how to successfully harness the power of partners. California Management Review, Volume 55, Issue 1 (2012): 24-46.
S28. Jansen, S., Cusumano, M., 2012. Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance. In Procedings of the 4th International Workshop on Software Ecosystems (IWSECO '12), pp. 41–58.
S29. Jansen, S., Finkelstein, A., Brinkkemper, S., 2009. A sense of community: A research agenda for software ecosystems. In Proceedings of the 31st Conference on Software Engineering, pp.187-190.
S30. Jansen, S.,Brinkkemper, S., Finkelstein, A., 2009. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. In proceedings of the 1st International Workshop on Software Ecosystems (IWSECO'09). pp. 34-48.
S31. Jansen, S., Brinkkemper, S., Souer, J., Luinenburg, L., 2013. The open software enterprise model: how open is my software business? In Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. UK: Edward Elgar Publishing, pp. 159-186.
S32. 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 System and Software. Volume 85, Issue 7, pp.1495-1510.
S33. Jansen, S., 2015. Opening the Ecosystem Flood Gates: Architecture Challenges of Opening Interfaces Within a Product Portfolio. In proceedings of European Conference on Software Architecture (ECSA'15), Springer International Publishing, pp. 121-136.
S34. Jansen, S., 2014. Measuring the health of open source software ecosystems: Beyond the scope of project health. Information and Software Technology, Volume, 56, Issue 11, pp. 1508-1519.
S35. Kapoor, R., Lee, J.M., 2013. Coordinating and competing in ecosystems: How organizational forms shape new technology investments. Strategic Management Journal, Volume 34, Issue 3, pp. 274-296.
S36. Kim, H.J., Kim, L., Lee, H., 2016. Third-party mobile app developers' continued participation in platform-centric ecosystems: An empirical investigation of two different mechanisms. International Journal of Information Management, Volume 36, Issue 1, pp. 44-59.
S37. Knauss, E., Damian, D., Knauss, A., Borici, A., 2014. Openness and requirements: Opportunities and tradeoffs in software ecosystems. In Proceedings of the IEEE 22nd International Requirements Engineering Conference (RE'14), 2014, pp.213-222.
S38. Koch, S., Kerschbaum, M., 2014. Joining a smartphone ecosystem: Application developers' motivations and decision criteria. Information and Software Technology, Volume 56, Issue 11, pp. 1423-1435.
S39. Manikas, K., Klaus, M.H., and Kyng, M,K., 2014. Governance mechanisms for healthcare apps. In Proceedings of the 2014 European Conference on Software Architecture Workshops. ACM, pp.10.
S40. Mizushima, K., Ikawa,Y., 2011. A structure of co-creation in an open source software ecosystem: A case study of the eclipse community. In Proceedings of

Technology Management in the Energy Smart World (PICMET'11). pp.1-8.
S41. Monteith, J.Y., McGregor, J.D., Ingram, J.E., 2014. Proposed metrics on ecosystem health. In Proceedings of the 2014 ACM international workshop on Software-defined ecosystems (BigSystem'14), ACM, pp.33-36.
S42. Mukhopadhyay, S., de Reuver, M., and Bouwman, H.,2016. Effectiveness of control mechanisms in mobile platform ecosystem. Telematics and Informatics, Volume 33, Issue 3, pp.848-859.
S43. Olsson, H., and Bosch, J., 2015. Strategic Ecosystem Management: A multi-case study on challenges and strategies for different ecosystem types. In proceedings of Software Engineering and Advanced Applications (SEAA'15), pp. 398-401.
S44. Pilinkienė, V., Mačiulis, P., 2014. Comparison of Different Ecosystem Analogies: The Main Economic Determinants and Levels of Impact. Procedia - Social and Behavioral Sciences, Volume 156, Issue 26, pp. 365-370.
S45. Popp, K.M., 2010. Goals of Software Vendors for Partner Ecosystems – A Practitioner ́s View. In proceedings of First International Conference on Software Business (ICSOB'10). Springer: pp.181-186.
S46. Sadi, M.H., Yu, E., 2015. Designing Software Ecosystems: How Can Modeling Techniques Help?In proceedings of Enterprise, Business-Process and Information Systems Modeling. Springer International Publishing, pp. 360-375.
S47. Sadi, H., Yu, E., 2014. Analyzing the evolution of software development: From creative chaos to software ecosystems. In Proceedings of the IEEE Eighth International Conference on Research Challenges in Information Science (RCIS'14), pp.1-11.
S48. Santos, R., Werner, C., 2012. Treating social dimension in software ecosystems through ReuseECOS approach. In Proceedings of the 6th IEEE International Conference on Digital Ecosystems Technologies (DEST'12), pp.1-6.
S49. Schönberger, A., Elsner, C., 2015. Modeling Partner Networks for Systematic Architecture Derivation in Software Ecosystems. In: Linnhoff-Popien, Claudia; Zaddach, Michael; Grahl, Andreas (Ed.): Marktplätze im Umbruch – Digital e Strategien für Services im Mobilen Internet.Springer Berlin Heidelberg , pp 655-665.
S50. Schultis, K.B., Elsner, C., Lohmann, D., 2014. Architecture challenges for internal software ecosystems: a large-scale industry case study. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'14), ACM, pp.542-552.
S51. Tiwana, A. Konsynski, B. Bush, A. 2010. Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. Information Systems Research. Volume 21, Issue 4, pp. 675–687.
S52. Valenca, G., Alves, C., Heimann, V., Jansen, S., Brinkkemper, S., 2014. Competition and collaboration in requirements engineering: A case study of an emerging software ecosystem. In Proceedings of the IEEE 22nd International Requirements Engineering Conference (RE'14), pp.384-393.
S53. van Angeren, J., Kabbedijk, K.,Popp, K., Jansen, S., 2013. Managing software ecosystems through partnering. In Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Cheltenham, UK: Edward Elgar Publishing, pp 85-102.
S54. van Angeren, J., Alves, C., Jansen, S., 2016. Can We Ask You To Collaborate? Analyzing App Developer Relationships in Commercial Platform Ecosystems. Journal of Systems and Software, Volume 113, pp. 430-445.
S55. van Angeren, J., Jansen, S., Brinkkemper, S., 2014. Exploring the relationship between partnership model participation and interfirm network structure: An analysis of the office365 ecosystem, In Proceedings of Software Business (ICSOB'14). Springer, pp. 1-15.
S56. Ven, K., Mannaert, H., 2008. Challenges and strategies in the use of Open Source Software by Independent Software Vendors. Information and Software Technology, Volume 50, Issues 9–10, pp.991-1002.
S57. Viljainen, M., Kauppinen., M., 2013. Framing management practices for keystones in platform ecosystems. In Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Cheltenham, UK: Edward Elgar Publishing, pp 121- 137.
S58. Viljainen, M., Kauppinen, M., 2011. Software Ecosystems: A Set of Management Practices for Platform Integrators in the Telecom Industry. In proceedings of 2nd International Conference on Software Business. Springer Berlin Heidelberg, pp.32-43.
S59. Wareham, J., Fox, P.B., CanoGiner, J.L., 2014. Technology ecosystem governance. Organization Science Volume 25, Issue 4, pp 1195-1215.
S60. West, J., and Wood, D., 2008. Creating and Evolving an Open Innovation Ecosystem: Lessons from Symbian Ltd. Available at SSRN Working Paper Series.
S61. West, J., and Wood, D., 2011. Tradeoffs of Open Innovation Platform Leadership: The Rise and Fall of Symbian Ltd. In proceedings of Social Science and Technology Seminar Series.
S62. Wnuk, K., Manikas, K., Runeson, P., Lantz, M., Weijden, O., Munir, H., 2014. Evaluating the governance model of hardware-centric software ecosystems: a case study of the axis ecosystem. In proceedings of 5th International Conference on Software Business, pp. 212-226.
S63. Yu, E., Deng, S., 2010. Understanding Software Ecosystems: A Strategic Modelling Approach. In International Workshop on Software Ecosystems (IWSECO'10).