Designing, Developing, and Implementing Software Ecosystems: Towards a Step-wise Guide.

Konstantinos Manikas[13], Mervi Hämäläinen[2], and Pasi Tyrväinen[2]

[1] Department of Computer Science
University of Copenhagen, Denmark
k@manikas.dk
[2] Agora Center
University of Jyväskylä, Finland
[Mervi.A.Hamalainen,Pasi.Tyrvainen]@jyu.fi
[3] DHI Group
Hørsholm, Denmark

Abstract. The notion of software ecosystems has been popular both in research and industry for more than a decade, but how software ecosystems are created still remains unclear. This becomes more of a challenge if one examines the "creation" of ecosystems that have high probability in surviving in the future, i.e. with respect to ecosystem health.
In this paper, we focus on the creation of software ecosystems and propose a process for designing, developing, and establishing software ecosystems based on three basic steps and a set of activities for each step. We note that software ecosystem research identifies that ecosystems typically emerge from either a company deciding to allow development on their product platform or from a successful open source project. In our study we add to this knowledge by demonstrating, through two case studies, that ecosystems can emerge from more than a technological infrastructure (platform). We identify that ecosystems can emerge out of two more distinct types of environments and thus the design should be based on the characteristics of this categorization.
Moreover, we follow the approach that design, development, and establishment are not three distinct phases but rather aspects of a single re-iterating phase and thus propose the view of design, development, and establishment as a continous process, running in parallel with and interrelated to the monitoring of the ecosystem evolution.

Key words: software ecosystems; software ecosystem design; software ecosystem health

1 Introduction

The notion of software ecosystems is argued to provide clear advantages compared to traditional software development and distribution as it, among other, accelerates software development, reduces time to market, and increases user and

customer segment reachability. It is not a surprise that within the recent years we have experienced an increasing popularity of software ecosystems both as a topic of study and as a means of developing and distributing software (products). Despite the popularity, it is still very challenging to *create* software ecosystems, especially if one should take into consideration aspects of ecosystem survivability, productivity, or health. Few studies have been investigating the conditions of establishment of a software ecosystems and even fewer propose ways of designing software ecosystems. However, this kind of studies tend to either be too specific for a type of ecosystem and thus hard to generalize, or too generic and thus hard to apply. Remarks that are already identified in the most recent and extensive systematic literature review [10], reviewing a total of 231 academic publications studying 129 software ecosystems.

Contemporary public discussion on software ecosystems is much driven by the most visible players in the digital economy, the platforms and app stores of Apple and Google being the usual examples in the discussion. Among the practitioners, this has lead to a platform-centric view of ecosystem thinking where a platform provider is needed to orchestrate an ecosystem. Further, the terms platform and ecosystem are closely connected if not treated almost as synonyms. However, the literature has presented a variety of ecosystems and value networks beyond the platform-centric approach, such as ecosystems build around standards, common business and commonly adopted infrastructure [8]

The limitations of platform-centric ecosystem thinking are, to some extent, visible also in our common thinking on how to build ecosystems. That is, we tend to think that the only way to build an ecosystem is to build a platform and attract participants to it by some means, typically by providing financial benefits to the participants. This underlying assumption may lead to ignorance of a wider view on how to build ecosystems as the viewpoints of actors in the value network and the value creation in the business domain are overlooked if not excluded totally from our thinking.

In this paper, we take the wider view to building ecosystems. We start our journey towards a method for building ecosystems from the observation that ecosystem can emerge out of three distinct types of environments and thus the design should be based on the characteristics of this categorization. We study two cases presenting an actor-rooted and a business-rooted approach to ecosystem building. Adding findings from the two cases to the infrastructure-rooted approach (including platform-centric approach) we propose a process for designing, developing, and establishing software ecosystems based on three basic steps and a set of activities for each step. Moreover, we follow the approach that design, development, and establishment are not three distinct phases, but rather aspects of a single re-iterating phase and thus propose the view of design, development, and establishment as a continuous process, running in parallel with and interrelated to the monitoring of the ecosystem evolution.

2 Background and related work

The field of software ecosystems has an activity that spreads through several years. From the first reference in the book of Messerschmitt and Szyperski [16] and the first publications in 2007, to the day, there have been several studies that have been examining software ecosystems as a whole and attempt to analyse, model, classify, or design software ecosystem. In this context Jansen et al. [7] proposed the analysis of software ecosystems from three perspective: software ecosystem level, software supply network level, and software vendor level.

Campbel and Ahmed [1] propose the analysis of software ecosystems into three components. Manikas and Hansen [14] analyse the literature of software ecosystems and identify, among other, a lack of consistency in what is a software ecosystem. They analyze the existing definitions and identify three main components: common software, business, and connecting relationships. Christensen et al. [3] propose the modellign and design of software ecosystems based on the concept of software ecosystem architecture consisted of three structures: organizational, business, and software. Knodel and Manikas [8] challenge the existing definition of software ecosystems and propose a set of building blocks for software ecosystems. Manikas and Hansen [13] focus on the concept of ecosystem health where they analyse the literature and propose a framework for defining ecosystem health. Hyrynsalmi et al [6] expand on this work to include 38 papers on health, while Hansen and Manikas [5], inspired by natural ecosystems, focus on defining the influence of individual actors to the ecosystem.

3 The cases

In this section we discuss and analyze two cases of designing and building a software ecosystem. The first case is the *telemedicine ecosystem* established around the telemedicine services of the Danish healthcare and the second cases is the *smart city ecosystem* established around the smart city and Internet of Things (IoT) infrastructure and services in an area of one of the most populated cities in Finland.

3.1 Telemedical ecosystem

Danish healthcare, following the tendency in many other western countries, is facing a number of challenges due to changes in the demographics. The increase in life expectancy and decrease of birth-rate in combination with a rapid increase of lifestyle conditions and the continuously improving healthcare diagnosis and treatment are putting a pressure on the economics of a welfare-based[1] and position the continuous care of the elderly and the chronically ill in even more central focus [9]. Telemedicine, comes as solution to these challenges. Telemedicine is understood as the provision of health through a distance. However, telemedical

[1] I.e. funded indirectly by collected tax.

technologies are faced with severe integration and interoperability issues caused by the increasing need to interact with other medical system characterized as "silo" solutions and organizationally complex systems [2]. The establishment of a software ecosystem comes to address these technical challenges and abstract the development of telemedical solutions from the resource-heavy task of integration and distribution.

Thus, the establishment of the telemedical ecosystem deviates from the typical view of ecosystem emergence (i.e. from a successful platform or product). In this ecosystem, the design was motivated from a set of clear incentives. The state and healthcare authorities have been part of shaping and clarifying the incentives, however this kind of actors have not been otherwise active in the design and establishment of the ecosystem. Therefore, the ecosystem was, during design, characterized form the lack of orchestration. The steps taken to establish the ecosystem was[2]:

- Identify and map the existing (and future) actors, (software) systems and their relationships [12].
- Identify the incentives for the different actors and make them explicit [3].
- Build the infrastructure that will support the ecosystem.

3.2 Smart city ecosystem

The second case is the establishment of an ecosystem in the *smart city* domain. The contribution of the digital technologies is considered to form a foundation for so called smart cities. Smart cities are complex systems and consist of multiple domains like transportation, energy, living, and governance. Smart city domains utilize digital technologies by collecting and storing both private and public data. They increasingly release the public information and data sets for external parties. The idea behind releasing the public data sets is to provide a possibility for external stakeholders to develop and create smart applications and services for citizens. Naturally, an ecosystem would support and facilitate the actor and smart city service interaction. An example is the environment for agile software and internet of things product and service development and experimentation with real users (citizens) in real-world settings [4].

In this context, our case, an urban area in one of the ten most populated cities in Finland is on the process of establishing a *smart city ecosystem*. The ecosystem establishment process was initiated by a set of actors interested in the smart city domain. These actors created a consortium that aimed at promoting the interaction of digital and software services in collaboration with independent business models, i.e. an ecosystem. Purpose of the smart city ecosystem is to develop new applications and internet of things service solutions in collaboration with construction companies, smart grid providers, nursing houses, city governance, and citizens. The initial actors in the smart city ecosystem included representatives from universities and city as well as the stakeholders from private

[2] A more detail description on this work can be found in [3, 9].

sector like the network service providers, telecommunications operators, smart locking service providers, and organizations in the privacy and digital identity domains. The citizens have central role in the smart city district. As an outcome of the smart city ecosystem, new applications and services are created to improve the quality of citizens' every-day life and enhance the research and value creation of modern digital technology services in smart city domain. The process of establishing the ecosystem included the following steps:

- Identify and map ecosystem (to-be) actors.
- Define business aspects: actor incentives, value propositions, customer segments, and revenue streams.
- Build technological infrastructure (e.g. platform) to support the ecosystem.

4 Proposed approach

As noted, the two cases studies are examples of ecosystem established by other than a common technological infrastructure (or platform). The telemedicine ecosystem is a business-rooted[3], while the smart city ecosystems is an actor rooted[4]. These two cases contribute with different perspectives on how ecosystem are established. They add more parameters to the up-to-now knowledge of ecosystems being created by a successful or popular technological infrastructure (platform) [10, 11, 14].

Up to the current point and to the best of our knowledge of the field, there is no previous work suggesting an applicable and holistic or generic (i.e. applicable to most or all types of ecosystems) way of creating a software ecosystem. This is the gap that we are trying to address with this approach, as we argue that a method for designing ecosystems that is easy to apply and mature enough would support the maturity of the field both theoretically and empirically.

In our approach, we propose the view of ecosystem design, development, and establishment as one continuous and re-iterative phase rather than three distinct phases. In order to initiate this process, the basic information needs to be collected and the first initial designs need to be drawn. Thus, we identify three main steps in our process to conduct the necessary work for the iterative design. Figure 1 shows the proposed steps and the tasks included in each step. Our approach includes three main steps: pre-analysis, design, and evaluate & monitor. In the subsections bellow we describe these steps. Our approach has a strong focus on the ecosystem health, thus apart from the design, we support the view of continuous monitoring and evolution of the ecosystem making the separation between design and establishment unclear. This is reflected in step 3.

Furthermore, taking the approach demonstrated from our cases, we identify that ecosystem design can occur based on three different ecosystem types: *infrastructure-rooted*, where the ecosystem is established around a technological

[3] I.e. initiated by strong actor incentives.
[4] I.e. initiated by a set of actors to drive the ecosystem development.

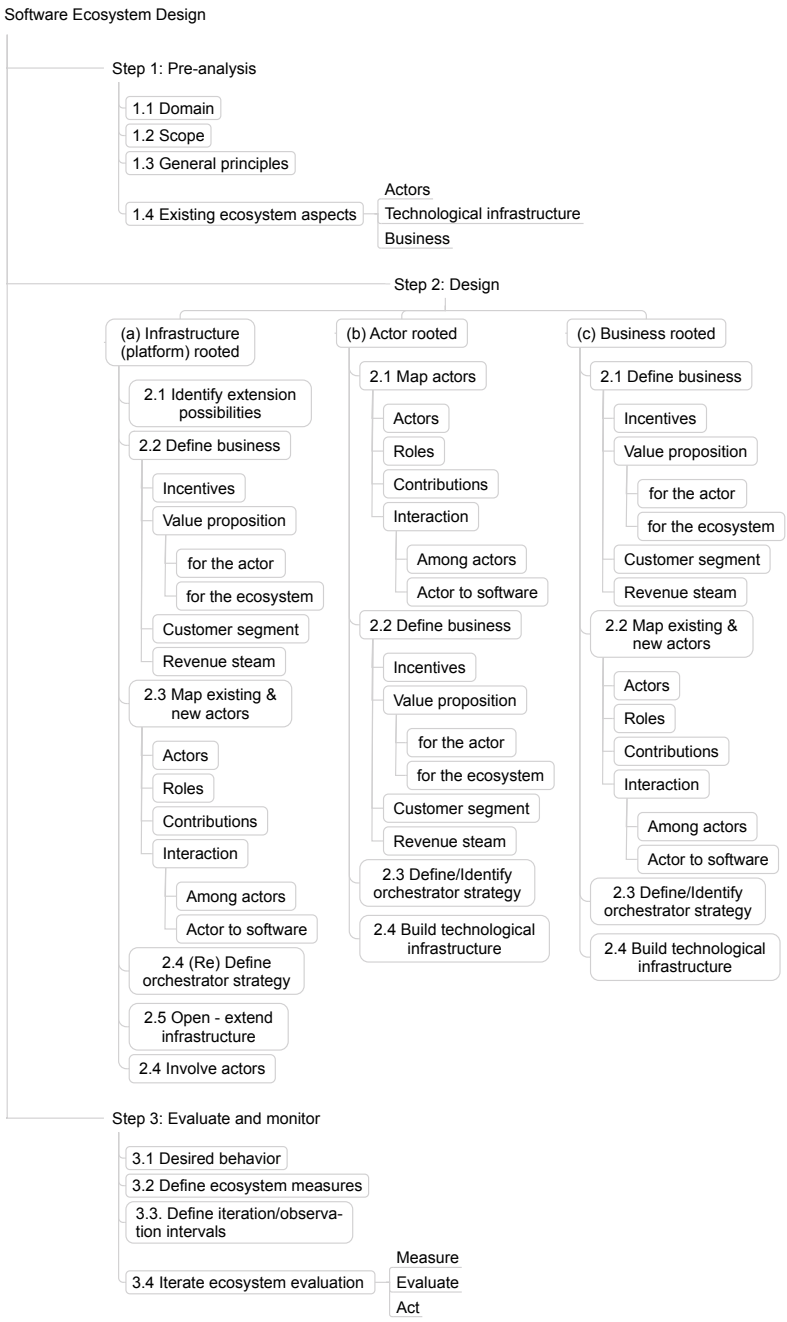

Fig. 1. Ecosystem design steps.

infrastructure[5];*actor-rooted*, where the establishment is around a strong actor consortium; and *business-rooted*, where the ecosystem is established around a strong business (or incentives).

4.1 Step 1: Pre-analysis

The initial step for the design of a software ecosystem is to identify the general information and characteristics of the future ecosystem. This includes identifying the applied domain of the ecosystem, i.e. how is the domain defined and what are the general characteristics of this domain. Further, this step includes defining the scope of the ecosystem and marking the borders of what is considered part of the ecosystem. Moreover, this step includes identifying the general *principles* of the ecosystem, i.e. core values and characteristics of the ecosystem that essential for the ecosystem [15]. Finally, part of the pre-analysis step includes identifying what aspects of the future ecosystem already exist that can form the base for the future ecosystem. This step will define whether the ecosystem is actor, infrastructure, or business rooted in step 2.

4.2 Step 2: Design

If we examine how ecosystems are created, the most common way appearing in the literature is from a (software) company opening their platform to external actors or an open source software (OSS) project that is gaining popularity. Examining the existing ecosystems in the industry (or in the literature e.g. the list in [10]), we note that this is not the only way that these ecosystem were established. Part of our proposed approach is to tailor the ecosystem design and establishment according to different aspects that exist in the domain of the future ecosystem. The above mentioned examples of OSS projects or companies opening the platform are examples of a *infrastructure rooted* ecosystems-to-be, since they have the base of what cold eventually become the common technological infrastructure of the future ecosystem. Another category is the *actor rooted*, that are ecosystems where there is a (strong) set or network of actors that can be form the core of the future ecosystem. Finally, there is also the *business rooted*, where there is a existing business potential and incentives (not necessarily for and from many actors) that can be the main drivers to the establishment of an ecosystem. An example of this can be found from the literature on evolution of vertical software industries where ecosystems emerge around new standards and platforms to enable effective collaboration between businesses/enterprises [17, 18].Clearly, the steps towards designing and establishing an ecosystem are different depending on the already existing aspects. Sub-steps (a),(b), and (c) list the actions for each type.

[5] Here using the approach of [8], we identify that an infrastructure can be apart from a platform, a standard or a protocol.

4.3 Step 3: Evaluate and monitor

Finally, as already explained, in our approach we propose the view of the design and development as a continuous and iterative process where software ecosystem design, development, and establishment are not distinct phases but rather part of one continuous and re-iterative phase. In order to achieve that, the ecosystem should be constantly monitored on its evolution and reaction to changes and potential deterioration should initiate new actions on the ecosystem architecture or orchestration. Thus, this step includes activities that focus on identifying what should be measured in the ecosystem to identify evolution and change in ecosystem health. After the measures are identified monitoring and evaluation activities will focus on (i) intervening in the operation of the ecosystem with changes and (ii) evaluating the effect of potential changes (as much as the whole design). It is essential to underline that identification of measures is an essential step as it defines the scope of action within the ecosystem. Too narrow measure might result in lack of overview of the whole ecosystem while not accurate or poorly defined measures might guide to wrong conclusions on the ecosystem activity and evolution.

5 Discussion

This paper aims at bringing focus to a central issue in the field of software ecosystem by proposing a method on designing ecosystems. Although generic and applicable, our method does not cover all the possible and potentially essential aspects in ecosystem design and evolution. One relevant aspect not adequately discussed is the *orchestration* of software ecosystems. The orchestration is central aspect in the health and evolution of an ecosystem and eventually the design of an ecosystem should include concrete considerations on the orchestration, in order to support the different characteristics of the ecosystem, its domain, and scope. Another relevant aspect is the establishment of the proper interfaces both technical and organizational. The different interfaces between the software components (e.g. in the common technological infrastructure) and between the different actors, should reflect the orchestration strategy of the ecosystem and respect the domain, boarders, and roles of the ecosystem and its actors.

Finally, as already discussed, the choice of the proper measures for monitoring the ecosystem is central to the evolution of the ecosystem towards the right direction. The monitored measures should also be influences as much as influence the ecosystem orchestration.

6 Conclusion and future work

In this work, we try to put focus on the gap in research and industry on how to "create" software ecosystems. Using our deep knowledge on software ecosystem literature and industry and experience from designing software ecosystem, we

propose a method for designing software ecosystems that is easy to use and applicable. Our method is consisted of three steps and a set of activities for each step.

We are currently empirically validating and improving this method. Further work includes the empirical evaluation and improvement with cases of each different type of design. Moreover, we plan to identify characteristics of the method for specific domains, i.e. how this "generic" method changes when applied to a domain with specific characteristics. It is our hope that this will be a first step towards a better informed and explicit design of software ecosystems and eventually further maturity in the field.

Acknowledgments

This work was partially conducted under the 5K[6] project, co-funded from the TEKES foundation (all authors), and the SCAUT[7] project, co-funded by Innovation Fund Denmark, grant #72-2014-1 (Manikas).

References

1. P. R. J. Campbell and F. 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.
2. H. B. Christensen, M. Christensen, K. M. Hansen, M. Kyng, K. Manikas, M. Surrow, and S. Urazimbetova. Requirements for a software-intensive ecosystem for telemedicine. *Med-e-Tel 2012*, pages 423–427, 2012.
3. H. B. Christensen, K. M. Hansen, M. Kyng, and K. Manikas. Analysis and design of software ecosystem architectures – towards the 4s telemedicine ecosystem. *Information and Software Technology*, 56(11):1476 – 1492, 2014.
4. M. Hamalainen and P. Tyrvainen. A framework for iot service experiment platforms in smart-city environments. In *2016 IEEE International Smart Cities Conference (ISC2)*, pages 1–8, Sept 2016.
5. K. M. Hansen and K. Manikas. Towards a Network Ecology of Software Ecosystems: an Analysis of two OSGi Ecosystems. In *Proceedings of the 25th International Conference on Software Engineering & Knowledge Engineering (SEKE'2013)*, pages 326–331, 2013.
6. S. Hyrynsalmi, M. Seppänen, T. Nokkala, A. Suominen, and A. Järvi. *Wealthy, Healthy and/or Happy — What does 'Ecosystem Health' Stand for?*, pages 272–287. Springer International Publishing, Cham, 2015.
7. 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.

[6] `https://agoracenter.jyu.fi/projects/5k`
[7] `http://www.scaut.dk/`

8. J. Knodel and K. Manikas. Towards a typification of software ecosystems. In J. M. Fernandes, R. J. Machado, and K. Wnuk, editors, *Software Business*, volume 210 of *Lecture Notes in Business Information Processing*, pages 60–65. Springer International Publishing, 2015.
9. K. Manikas. *Analyzing, Modelling, and Designing Software Ecosystems – Towards the Danish Telemedicine Software Ecosystem*. PhD thesis, Department of Computer Science, University of Copenhagen, Denmark, 2015.
10. K. Manikas. Revisiting software ecosystems research: A longitudinal literature study. *Systems and Software*, 117:84 – 103, 2016.
11. K. Manikas. *Supporting the Evolution of Research in Software Ecosystems: Reviewing the Empirical Literature*, pages 63–78. Springer International Publishing, Cham, 2016.
12. K. Manikas and K. M. Hansen. Characterizing the danish telemedicine ecosystem: Making sense of actor relationships. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems*, MEDES '13, pages 211–218, 2013.
13. K. Manikas and K. M. Hansen. Reviewing the health of software ecosystems - a conceptual framework proposal. In C. F. Alves, G. K. Hanssen, J. Bosch, and S. Jansen, editors, *Proceedings of the 5th International Workshop on Software Ecosystems, Potsdam, Germany, June 11, 2013*, volume 987, pages 33–44. CEUR-WS.org, 2013.
14. K. Manikas and K. M. Hansen. Software ecosystems – A systematic literature review. *Journal of Systems and Software*, 86(5):1294 – 1306, 2013.
15. K. Manikas, K. Wnuk, and A. Shollo. Defining decision making strategies in software ecosystem governance. Technical report, University of Copenhagen, Department of Computer Science, 2015.
16. D. Messerschmitt and C. Szyperski. Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books*, 1, 2003.
17. P. Tyrväinen. Vertical software industries. In *Vertical Software Industry Evolution*, chapter 2, pages 11–15. Physica-Verlag Heidelberg, 2009.
18. P. Tyrväinen, J. Warsta, and V. Seppänen. *Evolution of Secondary Software Businesses: Understanding Industry Dynamics*, pages 381–401. Springer US, Boston, MA, 2008.