Requirements Engineering for Innovative Software Ecosystems: A Research Preview

Karina Villela[1(✉)], Shashank Kedlaya[1,2], and Joerg Doerr[1]

[1] Fraunhofer IESE, Kaiserslautern, Germany
{karina.villela,joerg.doerr}@iese.fraunhofer.de
[2] TU Kaiserslautern, Kaiserslautern, Germany
skedlaya@rhrk.uni-kl.de

Abstract. [Context and motivation] In order to stay competitive in the Digital Transformation era, many organizations are engaging in innovative software ecosystems (SES). However, there is a lack of specific methods for tackling SES engineering challenges. [Question/problem] This paper presents a Requirements Engineering (RE) decision framework and a process for guiding key SES partners in the process of shaping their SES. [Principal ideas/results] Both the framework and the process build upon the results of a literature review and interviews with practitioners, and have undergone a preliminary qualitative evaluation. [Contribution] The systematic approach for shaping SES together with an explicit and clear definition of its application context will enable practitioners and researchers to apply it and/or translate it to other application contexts.

Keywords: Innovative digital solutions · Software ecosystems · Requirements Engineering · Software product management · Software platform management

1 Introduction

Digital transformation means "the profound and accelerating transformation of business activities, processes, competencies, and models aimed at fully leveraging the changes and opportunities of digital technologies and their impact across society in a strategic and prioritized way" [3]. The digital solutions in this scenario are inherently innovative, even disruptive. On the other hand, they integrate several complex and interdependent systems spanning multiple, interconnected application domains and provided by different organizations.

The paradigm of Software Ecosystems (SES) [5] can offer an answer to the challenge of developing the aforementioned digital solutions. However, Manikas's longitudinal literature study [9] concluded that there is a lack of specific theories, methods and tools for tackling SES problems. He argues that a big part of the problem derives from the fact that the notion of SES is very wide and arguably complex. Therefore, one of his recommendations is to focus more on research contributions to the field and explicitly characterize their context of application.

E. Knauss and M. Goedicke (Eds.): REFSQ 2019, LNCS 11412, pp. 117–123, 2019.
https://doi.org/10.1007/978-3-030-15538-4_8

In addition, Manikas's study [9] allows the conclusion that Requirements Engineering (RE) for Software Ecosystems (RE4SES) has been under-investigated.

Our organization has actively participated in several projects aimed at shaping innovative digital solutions based on SES (e.g., [11]). In these projects, the characteristic initial situation was the decision by some companies to combine their strengths in an SES in order to offer solutions that go far beyond their current and individual portfolio of solutions, towards the digital transformation of their business. This situation calls for a top-down approach, which means progressing from the definition of an overall SES concept towards innovative, typically more disruptive services, applications, and technical infrastructure. A bottom-up approach, i.e., gradually evolving the existing portfolio of solutions into a cohesive SES, would not achieve the desired level of innovation. Despite being typically applied in green field, top-down approaches to software engineering can also be applied in brown field. In this case, existing assets are incorporated if they fit the new SES concept. In this paper, we propose an RE decision framework and a process that build upon, but go beyond, our creativity workshops [12]. The goal is to provide holistic guidance to requirements engineers on how to contribute to the shaping of innovative SES as part of the SES leadership team [2].

Following Manikas's characterization scheme [9], our contribution is applicable to ecosystems whose orchestration is *not an anarchy*, whose means of value creation are *proprietary or hybrid*, and whose common technology can vary. Considering Bosch's category dimension and the spectrum between directed and undirected opening-up approaches [1], our contribution targets *applications* where the opening-up approach tends to be *directed*. We did not use Bosch's [1] platform dimension because ecosystems aiming at digital transformation frequently span all categories in this dimension.

The remainder of this paper is structured as follows: Sect. 2 presents our research method together with the findings from the literature and interviews with practitioners. Section 3 describes our RE decision framework and a process that sketches the dynamics among the decisions. In Sect. 4, we discuss the results of an initial qualitative evaluation of our contribution and present our future work.

2 Research Method and Main Findings

Our research has followed the five steps of the Design Science Research Cycle (DSRC) [15]: (1) awareness of the problem, (2) suggestion, (3) development, (4) evaluation and (5) conclusion. This research preview reports on the results from the first design cycle.

In the first step, we searched the literature for RE4SES challenges from the scientific perspective and performed individual, semi-structured 1-h interviews with ten practitioners from our organization in order to capture the practitioners' perspective. These practitioners reported on the RE challenges experienced in seven different projects regarding the conception and/or development of innovative SES. For the literature survey, we used the search string "Software Ecosystems" and ("Requirements" or "Requirements Engineering") in SpringerLink, ScienceDirect, IEEE Xplore, and ACM Digital Library. In the second step, we collected from the literature the currently available RE4SES approaches. Based on the knowledge acquired in steps (1) and (2), we designed an RE decision framework and a process, which will be described in Sect. 3. The evaluation step was performed by collecting feedback from eight practitioners (a subset of the previously interviewed ten practitioners, due to availability) in individual 1-h interviews, where the produced artifacts were explained to the interviewees, who were then asked to openly comment on their structure and contents and propose improvements.

Using a clustering approach over the literature survey's results, we extracted the following areas of challenges (introduced with an abbreviation for later reference):

- Requirements negotiation (***ReqNeg***) due to the interplay with several partners who must align their own interests and schedules and negotiate alternative solutions [10];
- Software integration (***SoftInt***) due to release planning cycles not being properly synchronized and product versions being launched at different points in time [17];
- Governance (***Gov***) due to the need to define clear responsibilities, make business strategy explicit, and determine the level of knowledge sharing [17];
- Support for the emergent requirements flow (***EmergFlow***) [7,14,17] due to the need to contextualize those requirements, map them to specific subsystems, and communicate them to the stakeholders [7].

The practitioners mentioned all these challenges and others: the uncertainty involved in the shaping of the ecosystem (***Unc***), as there is no one delivering concrete ecosystem requirements; the need to deal with several domains as well as with technical, legal, and business aspects simultaneously (***DomAsp***); the challenge of separating the requirements to be fulfilled by the platform (the common infrastructure in all our projects so far) and by the services and applications that will build on it (***PlatServ***); support for the on-boarding of new ecosystem partners (***OnB***), which requires the usage of prototypes and convincing capabilities; and change management (***CM***), due the need to deal with the inherent uncertainty and the on-boarding of partners.

As for the currently available RE4SES approaches, there are some approaches for dealing with specific challenges and activities [4,8,14,16]. However, there is still no guidance for requirements engineers on how to contribute to the top-down shaping of SES. As expected in competence/maturity models, Jansen et al. [6] present a wide range of capabilities, but at a very high level of abstraction. Santos and Werner [13] provide a set of concrete activities, but focus on the opening of existing platforms and on monitoring and management activities.

3 Decision Framework and Process for RE4SES

The proposed decision framework is composed of the following decision points:

Actors: organizations that interact or are expected to interact directly or indirectly with each other as part of the SES. Existing relationships such as trade relationships and collaborations are also of interest.

Business Strategy: flows that implement the ecosystem business, such as the flow of data, the flow of goods, and the flow of money.

Services/Applications: software services and applications that are required to implement the business strategy or influence it. Their identification is necessary to clarify which contributions are needed from the SES partners.

Openness Strategy: the degree of openness for the SES. This has two dimensions: (1) the ecosystem's openness for new partners, which can be tuned by entry conditions and facilities for the integration of contributions; and (2) openness of data, knowledge, artifacts and communication for the SES partners, which can be defined through IP rules, licensing policies, and collaboration principles.

Technical Infrastructure: requirements for the SES's common technological infrastructure at operation time and at development time, which are defined in alignment with the decisions about the openness strategy. This decision point also includes the identification of relevant data sources and the definition of user feedback mechanisms.

The decision points *Actors*, *Business Strategy* and *Service/Applications* were mainly derived from our workshop approach for the initial design of SES [12]; in addition, relationships among actors are addressed in [7,16]; the need to define clear responsibilities and make the business strategy explicit is mentioned in [7,17]; and Valença et al. refer to the identification of strategic features aimed at composing an SES roadmap [17]. The decision point *Openness Strategy* was motivated by [7] and [4]. The decision point *Technical Infrastructure* was also inspired by [7], where a technical infrastructure is made available to support decisions regarding openness, and by several references reporting the need to provide user feedback mechanisms [7,14,17].

For the application of the proposed decision framework, we envision the dynamics depicted in Fig. 1, which presents an iterative process that can be repeated until the SES concept is clear enough for the realization of its first version. According to Naab et al. [11], the first versions should cover a small subset of the SES focused on priority goals and on what needs to be solved in the short term, with the goal being to learn from the on-boarding of partners and the initial operation.

Preliminary Definition of SES Concept: Activities for defining the SES concept include the identification of actors and different end-user roles, the definition of the overall business strategy, the definition of software services and applications for composing the SES, and a preliminary discussion of openness alternatives. This also encompasses the indication by key SES partners of their intended contribution to the SES.

Elaboration of SES Enablers: Both the openness strategy and the technical infrastructure are key aspects for making the SES attractive to its current and

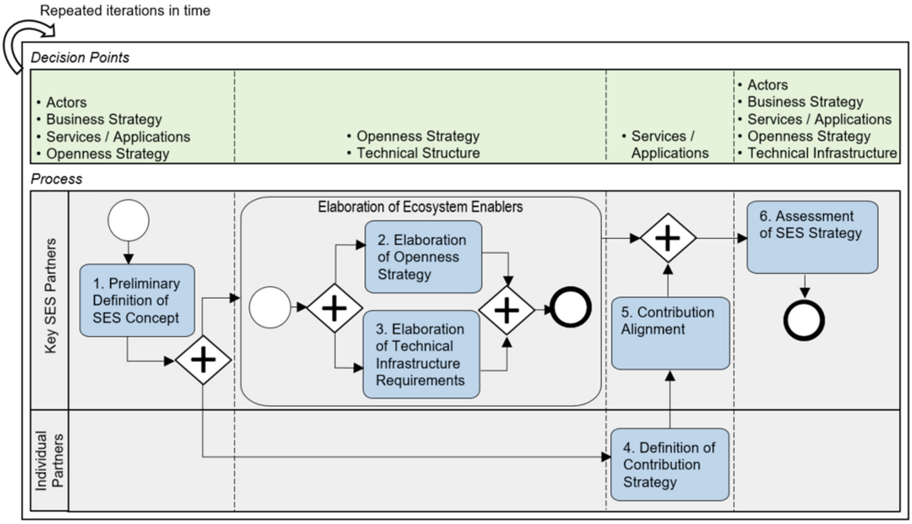

Fig. 1. Process for guiding the performance of RE4SES (BPMN 2.0)

potential partners. As the technical infrastructure goes beyond providing technical support to the openness strategy, the elaboration of the openness strategy and of the technical infrastructure requirements can start in parallel. The open software enterprise model [4] can help key SES partners in choosing the degree of openness for their SES.

Definition of the Contribution Strategy: Activities performed by each SES partner individually to determine their contribution to the SES. Linåker and Regnell [8] provide guidance on how to perform these activities. A contribution roadmap should indicate the features included in each contribution release as well as estimated release dates.

Alignment of Contribution Strategies: Activities to support the alignment of the contributions proposed by the SES partners, resulting in a joint SES roadmap. This potentially involves negotiation and may result in refinements of the overall SES concept and/or individual contributions. Knowing the power-dependence relations described in [16] is crucial for understanding power disputes and finding satisfactory solutions.

Assessment of SES Strategy: After making so many decisions and aligning individual contributions in a joint SES roadmap, it is time to assess the overall SES strategy in terms of the value it brings to all involved actors and end-users, consistency among all the decisions made, and the uncertainties and risks to be addressed and monitored from this time on.

The consideration of business, technical, and legal aspects [11] (*DomAsp*) is a concern that crosscuts all activities of the proposed process. The need to align the interests and contributions of the key SES partners [17] (derived from *ReqNeg* and *SoftInt*) is exactly the reason for: (1) having the key SES partners define the preliminary SES concept jointly (Fig. 1, Activity 1); (2) allowing the partners to define their contribution strategy separately (Fig. 1, Activity 4); and (3) giving the key SES partners the opportunity to jointly discuss and align their contributions (Fig. 1, Activity 5). This setting together with the activities in *Elaboration SES Enablers* (Fig. 1, Activities 2 and 3) addresses *Gov*. In addition, the aforementioned setting together with the possibility of having several iterations addresses *Unc* and makes the application of the process for shaping real-world SES realistic. We believe that *EmergFlow*, *OnB* and *CM* should be addressed in processes that are complementary to the proposed process. Finally, we intend to investigate how to refine our approach to address *PlatServ* explicitly.

4 Preliminary Evaluation and Future Work

The interviewees evaluated the full description of the decision framework and process. The evaluation was qualitative and open, with interviewees indicating the aspects that they particularly liked or disliked, and aspects that they missed. In some cases, they asked for information to be added to the description of the decision points. The critical improvement suggestions included making the SES platform more evident in the decision points and/or process, and addressing change management and the on-boarding activities. These improvement suggestions are exactly related to the practitioners' challenges that we believe should be addressed by complementary processes (*OnB* and *CM*) or that we want to address in future work (*PlatServ*).

The next step of this research will be to perform a thorough analysis of the evaluation interviews and thereby conclude the first design cycle. In the second design cycle, we will make the necessary adaptations to our contribution based on the received feedback. In addition, we will suggest notations for capturing decisions and propose the structure of the process artifacts. Regarding the evaluation in the second design cycle, we plan to carry out an industrial case study. In the long term, more case studies and other types of empirical studies should be performed in order to provide evidence of the suitability of the proposed approach and support its evolution.

Acknowledgement. This work was done in the context of the project Smart MaaS funded by the German Federal Ministry for Economic Affairs and Energy (grant number 01MD18014B).

References

1. Bosch, J.: From software product lines to software ecosystems. In: SPLC, pp. 111–119. Carnegie Mellon University (2009)
2. Hess, S., Knodel, J., Naab, M., Trapp, M.: Engineering roles for constructing ecosystems. In: ECSAW. ACM (2016). Article 24

3. i-SCOOP: Digitization, digitalization and digital transformation: the differences (2017). https://www.i-scoop.eu/digitization-digitalization-digital-transformation-disruption/. Accessed 22 Oct 2017
4. Jansen, S., Brinkkemper, S., et al.: Shades of gray: opening up a software producing organization with the open software enterprise model. J. Syst. Softw. **85**(7), 1495–1510 (2012)
5. Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: a research agenda for software ecosystems. In: ICSE, pp. 187–190. IEEE (2009)
6. Jansen, S., Peeters, S., Brinkkemper, S.: Software ecosystems: from software product management to software platform management. In: IW-LCSP@ICSOB, pp. 5–18 (2013)
7. Knauss, E., Yussuf, A., et al.: Continuous clarification and emergent requirements flows in open-commercial software ecosystems. RE J. **23**(1), 97–117 (2018)
8. Linåker, J., Regnell, B.: A contribution management framework for firms engaged in open source software ecosystems - a research preview. In: Grünbacher, P., Perini, A. (eds.) REFSQ 2017. LNCS, vol. 10153, pp. 50–57. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54045-0_4
9. Manikas, K.: Revisiting software ecosystems research: a longitudinal literature study. J. Syst. Softw. **117**, 84–103 (2016)
10. Manikas, K., Hansen, K.M.: Software ecosystems-a systematic literature review. J. Syst. Softw. **86**(5), 1294–1306 (2013)
11. Naab, M., Rost, D., Knodel, J.: Architecting a software-based ecosystem for the automotive aftermarket: an experience report. In: ICSA. IEEE (2018)
12. Nass, C., Trapp, M., Villela, K.: Tangible design for software ecosystem with Playmobil®. In: NordiCHI, pp. 856–861. ACM (2018)
13. Santos, R., Werner, C.: ReuseECOS: an approach to support global software development through software ecosystems. In: ICGSEW, pp. 60–65. IEEE (2012)
14. Schneider, K., Meyer, S., Peters, M., Schliephacke, F., Mörschbach, J., Aguirre, L.: Feedback in context: supporting the evolution of IT-ecosystems. In: Ali Babar, M., Vierimaa, M., Oivo, M. (eds.) PROFES 2010. LNCS, vol. 6156, pp. 191–205. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13792-1_16
15. Takeda, H., Veerkamp, P., Yoshikawa, H.: Modeling design process. AI Mag. **11**(4), 37 (1990)
16. Valença, G., Alves, C.: A theory of power in emerging software ecosystems formed by small-to-medium enterprises. J. Syst. Softw. **134**, 76–104 (2017)
17. Valença, G., Alves, C., et al.: Competition and collaboration in requirements engineering: a case study of an emerging software ecosystem. In: RE Conference, pp. 384–393. IEEE (2014)