



Defining Decision Making Strategies in Software Ecosystem Governance

Konstantinos Manikas
Krzysztof Wnuk
Arisa Shollo

Technical Report no. 2015/01

ISSN: 0107-8283

Defining Decision Making Strategies in Software Ecosystem Governance

Konstantinos Manikas
Department of Computer
Science (DIKU)
University of Copenhagen
Denmark
kmanikas@di.ku.dk

Krzysztof Wnuk
Department of Computer
Science
Lund University
Sweden
Krzysztof.Wnuk@cs.lth.se

Arisa Shollo
Department of Operations
Management
Copenhagen Business School
Denmark
ash.om@cbs.dk

ABSTRACT

Making the right decisions is an essential part of software ecosystem governance. Decisions related to the governance of a software ecosystem can influence the health of the ecosystem and can result in fostering the success or greatly contributing to the failure of the ecosystem. However, very few studies touch upon the decision making of software ecosystem governance. In this paper, we propose decomposing software ecosystem governance into three activities: input or data collection, decision making, and applying actions.

We focus on the decision making activity of software ecosystem governance and review related literature consisted of software ecosystem governance, organizational decision making, and IT governance. Based on the identified studies, we propose a framework for defining the decision making strategies in the governance of software ecosystems. We identify five decision areas for software ecosystem governance and four archetypes describing the way decisions are taken for each decision area. We explain this matrix-based framework by providing examples from existing software ecosystems.

Categories and Subject Descriptors

K.6.3 [Management of Computing and Information Systems]: Software Management—*Software development*; H.1.1 [Systems and Information Theory]: [General systems theory]; K.6.3 [Software Management]: Software development

General Terms

Theory; Design; Management

Keywords

decision making, software ecosystem governance, software ecosystems

1. INTRODUCTION

Software ecosystems can be very different ranging from e.g. non-profit, community-driven, open for anyone to participate to for-profit, with strict rules on who participates, of proprietary source. This variability can make the study of software ecosystem governance hard. Studying the ecosystem governance can give insights to elements and strategies that constitute a successful and sustainable ecosystem, i.e. a healthy ecosystem.

In this paper we analyze the governance of software ecosystems into three governance activities (as explained in Section 3): (i) input - data collection, where the set of information related to the ecosystem governance is collected, (ii) decision making, where the collected information is processed and a set of actions among a set of alternatives is chosen, and (iii) applying actions, where the decided actions are applied to the ecosystem. In continuation, we focus on the process of making decisions (activity (ii)) in the governance of a software ecosystems and identify it as an important aspect of the ecosystem influencing the (proper) applicability of the ecosystem strategy and governance, affecting the ecosystem health and functioning as a whole. Thus, we propose the study of the governance of software ecosystems through the building of a theoretical model that can provide possibilities to the research community to expand and empirically verify it. To do so, we review the existing software ecosystem governance literature along with the organizational decision making and IT governance and provide a framework for defining the decision making strategies.

Our work serves as means of setting focus on the decision making as an important activity in the governance of ecosystem. This includes making the decision making strategies explicit for software ecosystems, ensuring the alignment of governance aims and decision making strategies, and providing the background for the evolution in the study of software ecosystem governance.

In the following sections we explain the method used for this study (Section 2), provide the background of our study (Section 3), suggest a framework for the decision making in software ecosystems and classify the ecosystems from the literature (Section 4) and conclude our paper (Section 5).

2. METHOD

In this paper, we propose the analysis of ecosystem governance into three main activities whereas decision making is

one of them (see Figure 1). Thus, we propose the study of decision making as means of improving the quality governance and eventually improve the ecosystem health. According to the literature reviews of [19, 3] and additional literature database searches, decision making in software ecosystem governance has not been directly addressed in the literature. Therefore, in this study we examine the governance of software ecosystems in general and expand our literature scope to include fields that study decision making: organizational decision making and IT governance. We extract the identified literature from a set of published mapping studies and literature reviews for each of the fields in parallel with our own knowledge of the fields. For governance in software ecosystems, we use the studies of [19, 3], for the organizational decision making we use the reviews of [24, 20], and for IT strategy we use the review of [6]. The collected literature builds the background of our study and serves as input to our suggested input. We identify the main decision areas the governance of software ecosystem inspired from the ecosystem governance literature and propose a framework for analyzing the decision making models used in an ecosystem based on the background. In continuation we explain the framework by providing examples of existing software ecosystems.

3. BACKGROUND

In order to analyze the governance of software ecosystems, we propose the decomposition of governance into three main activities: (i) *input - data collection*, (ii) *decision making*, and (iii) *applying actions*. As shown in Figure 1, these activities are circular with each activity providing input to the next one. In a typical software ecosystem, governance scenario, the governing actors rely on one or more measures and information sources to collect data for the functioning of the ecosystem, or an aspect related to the ecosystem (activity (i)). The collected data serve as input to the decision making structure(s) where data are processed and interpreted to scenarios, the consequences and the probability of these scenarios are assessed, and possible actions towards these scenarios and the consequences of the actions are identified and evaluated in order to conclude to on or more actions (activity (ii)). In continuation, the concluded actions are applied to the ecosystem (activity (iii)). After applying the selected actions, feedback is collected to evaluate the choice of actions and thus start a new governance cycle.

The efficiency of the governance of an ecosystem depends on the quality of each of the activities: The quality of the input activity (i) depends on the extent the data collected are correct, accurate and represent the complete picture. The quality of the decision making activities (ii) depends on the correct processing and scenario translation of the input data and the identification and evaluation of the complete set of actions and consequences. While the quality of the applying action activity (iii) relies on the proper appliance of the actions in regards to the affected parties in the ecosystem and the consequences of the actions.

In this paper, we draw focus on the decision making of software ecosystem governance (activity (ii)) and propose the modeling of the decision making strategies in the governance of software ecosystems as means of setting focus on the decision making as an important activity in the governance,

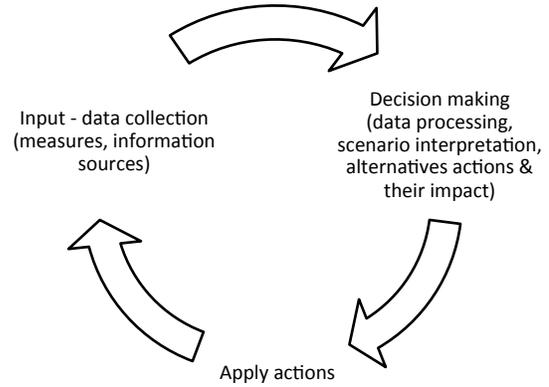


Figure 1: Software ecosystem governance activities.

making the decision making strategies explicit for software ecosystems, ensuring the alignment of governance aims and decision making strategies, and providing the background for the evolution in the study of software ecosystem governance.

3.1 Software ecosystems

When looking at the literature on software ecosystems, we note that decision making in ecosystem governance has not been addressed [19, 3]. However there is a number of studies in software ecosystem governance, that are related to decision making. Baars and Jansen [2] propose a framework to analyze the governance of software ecosystems for individual companies. The framework is consisted of five categories: explicitness of the ecosystem, explicitness of the governance, responsibility, measurement and knowledge sharing.

Jansen et al. [14] propose a model for measuring the degree of openness of a software producing organization. The model lists five areas where the organization can open up examined under three levels: strategic, tactical and operational. One of these areas is governance, where different options for opening up the governance are discussed for the three levels. Jansen and Cusumano [15], based on the two previous papers extract four classification factors for software ecosystems: underpinning technology, (type of) coordinators, extension market and accessibility (of the ecosystem). In continuation, they propose a governance model for the prevention and improvement of the ecosystem health. In this model they distinguish between a software (service) platform and a standard as the types of underpinning technology and a community or private entity for each technology. For the two separations (technologies and community/private entity), they propose actions that address each of the Iansiti and Levien health measures: niche creation, productivity, robustness [12]. Wnuk et al. [33] apply the above governance model to the Axis Application Development Partner (ADP) ecosystem, a hardware-centric ecosystem with product distribution particularities, and conclude that although the above model is useful in characterizing the governance of ADP, the model needs to be evolve to capture particular aspects of ecosystems like ADP.

Christensen et al. [7] model software ecosystems according to the concept of “software ecosystem architecture” and pro-

pose three structures, one of them being the *organizational structure* defined as the structure responsible for the ecosystem governance, i.e.. the structure that “contains actor and software elements that are related to the governance of the interaction and organization of the elements in the ecosystem”. They identify that the important aspects in the organizational structure are: the sets of actor and software elements, the ecosystem boundary, and the interactions and coordination among actors and software elements.

From the above we note two main points: Firstly, software ecosystem literature does not cover decision making while, as we explained in the previous section, we consider it an essential part of ecosystem governance. Secondly, the existing ecosystem governance models¹ have an operational perspective providing direct strategy suggestions. In our view software ecosystems can vary greatly in their structures, (e.g. organizational, actor, and software, thus we propose the study of ecosystem governance from an abstract model based on theoretical frameworks rather than empirical deduction.

3.2 Organizational decision making

An organization can be defined as “a consciously coordinated social entity, with a relatively identifiable boundary, that functions on a relatively continuous basis to achieve a common goal or a set of goals” [22]. Organizations are also seen as systems aimed at processing information [21, 28] with one of the purpose of information processing is to make decisions. Decision making in organizations, is a field that has attracted studies from the 1940’s. Shollo [24] provides an overview of the literature in organizational decision making. The basic distinction of decisions is between structured or programmed and unstructured or non-programmed decisions [24, 26]. Structured decisions include the use of criteria that are well-defined and measurable while unstructured are more focused on “problem solving” [26]. Simon makes the separation between operational decisions, that tend to be structured and strategic decisions, that tend to be unstructured. Weick [28] proposes the perspective of organizations as information processing systems for reducing information ambiguity.

Rationality in decision making or the rational choice is a concept that is as old as studies on human behavior [20]. March [20] defines rational as the procedure of making a decision (choice) based on four questions: *Alternatives*: “what actions are possible?”, *expectations*: “what consequences might follow and how likely these consequences are?”, *preferences*: “how valuable are the consequences of each choice?”, and *decision rule*: “how is a choice to be made among alternatives?”. Literature shows, however, that people, when making strategic decisions, do not base their choices on rational [25, 20, 8, 17], but use other means like judgment, intuition and negotiation [24, 17, 16, 9, 4].

Thompson [27] identifies two parameters in decision making: *goals* and *means* and identified three distinct possibilities: when the goals are clear and the means are unclear, when the goals are unclear and the means are clear, and when

both goals and means are unclear.

3.2.1 IT governance

When we look at the IT governance literature, we recognize that decision making plays an important role. Decision making appears as IT decision rights [30], decision loci of control (centralized and decentralized IT governance) [11] and expanded decision making structures studying the bipolar governance systems between management and IT [5]. In this study we are inspired from the work of Weill and Ross [30, 31] that is considered to be the extension of the two prevailing IT governance streams, linking the previous main evolution tracks of IT strategy together [6].

IT governance is mainly focusing on the separation between the business and the IT in a corporation. In this concept, IT is used as means of optimizing the business. Business strategies are not necessarily reflected in the IT decisions. Weil and Ross [30] proposed a framework that models the IT decisions. It is a matrix with five decision domains: IT principles, IT architecture, IT infrastructure strategies, business application needs and IT investments. For each of these domains, the input rights (how is information collected) and decision rights (how are decisions taken) are categorized in six politically inspired archetypes: ² (i) business monarchy where decisions are taken by the business corpus (CxOs), (ii) IT monarchy where IT professionals in the corporation make the decisions, (iii) feudal where decision are taken by autonomous business units, (iv) federal where decisions are made from a body of representatives that can be consisted of business units, business processes and IT, (v) IT duopoly where decisions are made from a joint body of IT and at least one business unit and (vi) anarchy.

We identify that IT governance and software ecosystem governance have a wide intersection, however, in order to use IT governance theories in software ecosystems we have to acknowledge the following differences:

IT - product separation. In IT governance there is a separation between the IT and the product where in software ecosystem the software is/can be the product itself. The decision about implementing a feature in an ecosystem may have serious implication on the entire IT infrastructure, for example a cloud storage features in the product based on an ecosystem. Therefore, we advocate that the separation between IT and ecosystems is in this case not possible.

IT - Platform. Information technology for IT governance can be parallelized with the ecosystem platform and other infrastructure. Corporations are using IT to increase productivity and efficiency and produce better and more antagonistic products but also to apply governance processes and strategies. The same functions are represented by the technological platform and possibly other technical infrastructures in software ecosystems.

Business versus IT and company versus community. IT strategy is focusing on the alignment of the business and the IT in the governance of organizations, while software

¹With the exception of [7] that does not solely focus on modeling ecosystem governance.

²The list of archetypes is from [29]. Variation of this list appear in the [30, 31, 32].

ecosystems are focusing on the alignment between (private) companies and communities.

4. DECISION MAKING FRAMEWORK

In the following section, we propose a framework for defining the decision making strategies applied to an ecosystem for the ecosystem governance. The purpose of this framework is to make explicit and analyze the decision making strategies used in the governance of software ecosystems. The analysis and explicitness of decision making strategies could assist in providing a clear overall governance strategy and uncover possible issues in the existing ecosystem governance actions. For example, if the ecosystem is having issues in a specific decision area, identifying how decisions are made in that area and eventually what actors are responsible for making these decisions could lead to the input model and action model they are using. This would be the first step to address the issue and thus improve the health of the ecosystem.

The ecosystem governance decision making framework consists of five main decision areas that group the main governance decisions of the ecosystem and four archetypes describing how decisions are taken for each of the areas. The framework with examples for existing ecosystems can be seen on Table 1. The main decision areas are:

Principles: Decisions under this area address the core principles, general values and main directions of the ecosystem. This area determines the general ecosystem strategy. Decisions here, usually have great influence and are fundamental to the whole ecosystem.

Actor Interaction: This area covers decisions related to and affecting the actors of the ecosystem. It might cover decisions on the total number of actors or other characteristics the actors of the ecosystem should have (e.g. how the ecosystem introduces new actors - openness, accessibility), or how the actors interact with each other, e.g. interaction, coordination, and rules.

Software Interaction: Decisions regarding the software interaction and structure of the software component network of the ecosystem. Decisions and actions related to this area include software release management, software architecture of the ecosystem, and other procedures and rules affecting the software build and distributed.

Platform: Decisions regarding the technological platform and other common technical infrastructure. Although the technological platform is a software component itself and affects the software interaction, we examine it as a separate decision area since it is a central and important part of the ecosystem coordinating both actor and software interaction to a big extent and providing the possibility to the orchestrators to apply governance actions through it, thus having a direct influence on the ecosystem's health [18]. Decisions on this area include the management of the platform, platform architecture, commercialization of the platform and platform openness (to what extent different actors can be involved in the core development of the platform). Product decision makers need to reason about these attributes when selecting the appropriate platform characteristics for the realization of their product features and product strategy.

SECO Business and Products: Decisions concerning the business models of the ecosystem, motivation of the actors and distribution and availability of products. E.g. app-stores strategies, vendor/reseller/value-added reseller strategies, actor involvement and incentives.

For each of the main decision areas, the way decisions are made is characterized by the archetypes bellow:

Monarchy: One actor alone making decisions for a specific decision area. E.g. Apple is the main hardware supplier and orchestrator of the App Store ecosystem, therefore is deciding on SECO principles decisions.

Collective: Where decisions are made through processes involving all the actors, e.g. voting. For example in the Django framework the developers are asked to vote for what new features should be implemented [10].

Federal: Where a number of actors are assigned as representatives to take decisions. This is the case with the Apache ecosystem where changes in the Apache server repository (platform) can be done from a specific group of actors, the committers. For normal developers to become committers, they are voted according to their contribution[1]. The same for the Open Design Alliance (ODA) where the founding members can decide to change the way new actors enter the ecosystem [14].

Anarchy: Where each actor decides on its own. An example is the World of Warcraft [15] where any actor can create their own app store or the Ruby on Rails ecosystem where anyone can commit to the platform [23].

We note that the decision areas are quite general and abstract and thus an ecosystem might have more than one ways of taking decisions (archetypes) for a decision area. For example an ecosystem deciding as a monarchy for a group of actors (possibly with activity of strategic importance) and as a collective or anarchy for another group of actors. The main decision areas might eventually be broken down to subcategories according to the specific parameters of the ecosystem.

5. CONCLUSION

The way a software ecosystem is governed affects the functioning of the ecosystem. Proper governance can increase the health and prosperity of the ecosystem and provide more value for the ecosystem actors. Decision making is an important part of the ecosystem governance: ensuring the explicitness of the decision making in the ecosystem governance and the alignment of the decision making processes with the general ecosystem strategies is an important factor that constitutes the proper ecosystem governance. The literature on the governance of software ecosystems provides us with a number of frameworks for studying ecosystem governance, but do not address the aspect of decision making.

In this study, we use the existing literature of software ecosystem governance and IT governance to create a framework for defining the way governance decisions are made in software ecosystems. We identify five decision areas for software ecosystem governance and four archetypes describing the way decisions are taken for each decision area. We explain

	Principles	Actor Interaction	Software Interaction	Platform	Business & Products
Monarchy	Apple AppStore				
Collective		ODA	Django		
Federal				Apache	
Anarchy			Ruby on Rails	World of Warcraft	

Table 1: The decision making framework for SECO governance

this matrix-based framework by providing examples from existing software ecosystems. The purpose of this framework is to bring focus, make explicit, and analyze the decision making strategies used in the governance of software ecosystems.

Acknowledgements

This work has been partially conducted under the Connect2Care project funded by the UNIK Partnership³ and partly by the SYNERGIES project.

6. REFERENCES

- [1] Apache Foundation. Apache community development – contributing to the apache software foundation. <https://community.apache.org/contributors/>. Accessed June 2014, 2014.
- [2] A. Baars and S. Jansen. A framework for software ecosystem governance. In M. A. Cusumano, B. Iyer, and N. Venkatraman, editors, *Software Business*, volume 114 of *Lecture Notes in Business Information Processing*, pages 168–180. Springer Berlin Heidelberg, 2012.
- [3] O. Barbosa and C. Alves. A systematic mapping study on software ecosystems. In *Third International Workshop on Software Ecosystems (IWSECO-2011)*, pages 15–26. CEUR-WS, 2011.
- [4] M. Bazerman and D. A. Moore. *Judgment in managerial decision making*. Wiley, 2008.
- [5] A. C. Boynton and R. W. Zmud. Information technology planning in the 1990’s: Directions for practice and research. *MIS quarterly*, 11(1):59–71, 1987.
- [6] A. Brown and G. Grant. Framing the frameworks: A review of it governance research. *Communications of the Association for Information Systems (Volume 15, 2005)*, 696(712):712, 2005.
- [7] 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*, 0(0):–, 2014.
- [8] R. M. Cyert, J. G. March, et al. A behavioral theory of the firm. In J. B. Miner, editor, *Organizational Behavior 2: Essential theories of process and structure*, volume 2. Englewood Cliffs, NJ, 1963.
- [9] E. Dane and M. G. Pratt. Exploring intuition and its role in managerial decision making. *Academy of Management Review*, 32(1):33–54, 2007.
- [10] Django framework. Reporting bugs and requesting features | django documentation | django. <https://docs.djangoproject.com/en/dev/internals/contributing/bugs-and-features/>. Accessed June 2014, 2014.
- [11] J. T. Garrity. Top management and computer profits. *Harvard Business Review*, 41(4):6–12, 1963.
- [12] M. Iansiti and R. Levien. Strategy as ecology. *Harvard Business Review*, 82(3):68–81, 2004.
- [13] S. Jansen, S. Brinkkemper, and M. A. Cusumano, editors. *Software Ecosystems – Analyzing and Managing Business Networks in the Software Industry*. Edward Elgar, Cheltenham, UK, 2013.
- [14] S. Jansen, S. Brinkkemper, J. Souer, and L. Luinenburg. Shades of gray: Opening up a software producing organization with the open software enterprise model. *Journal of Systems and Software*, 85(7):1495 – 1510, 2012.
- [15] S. Jansen and M. A. Cusumano. Software ecosystems – analyzing and managing business networks in the software industry. In Jansen et al. [13], chapter Defining software ecosystems: a survey of software platforms and business network governance, pages 13–28.
- [16] G. A. Klein, J. E. Orasanu, R. E. Calderwood, and C. E. Zsombok. Decision making in action: Models and methods. In *This book is an outcome of a workshop held in Dayton, OH, Sep 25–27, 1989*. Ablex Publishing, 1993.
- [17] A. Langley, H. Mintzberg, P. Pitcher, E. Posada, and J. Saint-Macary. Opening up decision making: The view from the black stool. *organization Science*, 6(3):260–279, 1995.
- [18] 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.
- [19] K. Manikas and K. M. Hansen. Software ecosystems – A systematic literature review. *Journal of Systems and Software*, 86(5):1294 – 1306, 2013.
- [20] J. G. March. *A primer on decision making - How decisions happen*. Free Press, 19994.
- [21] J. G. March and H. A. Simon. *Organizations*. Wiley, 1958.
- [22] S. P. Robbins. *Organization Theory: Structures, Designs, And Applications*. Prentice Hall, 1990.
- [23] Ruby on Rails. Contributing to ruby on rails - ruby on rails guides. http://guides.rubyonrails.org/contributing_to_ruby_on_rails.html. Accessed June 2014, 2014.
- [24] A. Shollo. *The Role of Business Intelligence in*

³<http://www.partnerskabetunik.dk/>

Organizational Decision-making. PhD Series.
Copenhagen Business School [Phd], 2013.

- [25] H. Simon. *Administrative behavior; a study of decision-making processes in administrative organization*. Macmillan, 1947.
- [26] H. A. Simon. *The New Science of Management Decision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1977.
- [27] J. Thompson. *Organizations in action : social science bases of administrative theory*. McGraw-Hill, 1967.
- [28] K. Weick. *The Social Psychology of Organizing*. Topics in social psychology. Random House, 1979.
- [29] P. Weill. Don't just lead, govern: How top-performing firms govern it. *MIS Quarterly Executive*, 3(1):1–17, 2004.
- [30] P. Weill and J. W. Ross. *IT-Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business Press, 2004.
- [31] P. Weill and J. W. Ross. A matrixed approach to designing IT governance. *MIT Sloan Management Review*, 46(2):26–34, 2005.
- [32] P. Weill and R. Woodham. Don't just lead, govern: Implementing effective it governance. *MIT Sloan Management Review*, 2002.
- [33] K. Wnuk, K. Manikas, P. Runeson, M. Lantz, O. Weijden, and H. Munir. Evaluating the governance model of hardware-centric software ecosystems - a case study of the axis ecosystem. In *Software Business (ICSOB 2014). 5th International Conference on*, pages –, 2014.