Please quote as: Dellermann, D.; Jud, C.; Lipusch, N. & Popp, K. M. (2018): Managing the Dark Side of Software Ecosystems. In: Social Science Research Network (SSRN).

Managing the Dark Side of Software Ecosystems

Dominik Dellermann

University of Kassel, Pfannkuchstrasse 1, 34119 Kassel, Germany. E-mail: dellermann@uni-kassel.de

Christopher Jud

University of Stuttgart, Chair for Information Systems, Stuttgart, Germany. E-mail: jud@wius.bwi.uni-stuttgart.de

Nikolaus Lipusch

University of Kassel, Pfannkuchstrasse 1, 34119 Kassel, Germany. E-Mail: nikolaus.lipusch@uni-kassel.de

Karl Michael Popp

SAP SE, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany. E-Mail: karl.michael.popp@sap.com

Abstract

The emergence of platforms is significantly changing the organizing logic of software development. Platform owners are increasingly engaging vibrant ecosystems around their platform to foster third-party innovation. Despite all the potential benefits for complementors, however, innovation in platform ecosystems also introduced essential new risks that remain under investigated. To discover the complementors´ risk in software ecosystems we utilize 21 interviews and 42 survey respondents to examine risk and governance mechanisms such as control structures and architectural choices to handle such.

1. The emergence of software ecosystems and complementors

The emergence of technological platforms like Salesforce´s Force.com, SAP´s HANA or Apple's iPhone operating system (iOS) substantially changed the logic of value creation in the software industry [1]. While the traditional business logic has been the independent development of a monolithic product by an individual software vendor, modern software strongly relies on value added services from third-party developers, called complementors [2]. Modular cloud platform architecture enables complementors to develop own applications independently using capabilities of the platform, i.e. an expandable code base [3]. The modularity of the platform and its modules attempts to minimize interdependence between both by decoupling the artifacts and make use of standardized interfaces [3,4]. Decoupling allows that changes within a module do not require parallel changes in the platform and vice versa. Standardization refers to the use of application programming interfaces (APIs) that are applied to meet conformance between the platform and the applications [5]. The interfaces of the platform allow deployment plugging in, and the development of complementary innovation, like software applications and services [6]. Often, APIs are defined by the platform vendor or by suppliers in coordination with the platform vendor.

The platform and its corresponding modules form an ecosystem in which numerous participants, including the platform vendor (e.g. Salesforce.com, SAP), suppliers, end users, and complementors, transact with one another in complex ways to develop novel value propositions for end users [7]. Such ecosystems are a set of actors functioning as a unit and interacting with a shared market for products and services bond together with multiple interaction relationships among them [8]. These relationships are underpinned by a common technological platform and operate through the exchange of data, information, resources and artifacts [3]. From a network perspective, the platform vendor is referred to as central "hub" that maintains "loosely coupled" relationships with ecosystem participants that develop third-party innovations [9,10]. Platform vendors are therefore increasingly engaged in facilitating vibrant ecosystems and fostering innovation on their digital platforms as the number of complementary applications increases not just the functionality but also the overall value of a platform [1]. Such ecosystems are becoming the dominant form of organizing software development in various domains, for instance enterprise software (EAS), mobile, or the Internet of Things (IoT).

To shift design capability to complementors, platform vendors offer resources that enable complementors to join the ecosystem and to offer their products to end-users. Such platform boundary resources are technical resources like software development kit (SDK) and or APIs as well as social boundary resources, such as monetary and non-monetary incentives or intellectual property (IP) rights.

2. The complementors´ risk in software ecosystems

Despite all the potential benefits for complementors, however, platform ecosystems have also been known for its fluctuation and high rates of desertion [11]. The accelerating dependence on the platform vendor on a technical, contractual or market access level, however, has not only created new business opportunities but also introduced essential new risks that emerge through the loosely coupled architecture. These risks go beyond traditional risks of software projects [12] because innovation on platforms has blurred conventional firm boundaries and includes multiple and heterogeneous organizations. The specific characteristics of software platforms and ecosystems create also significant changes in the nature and analysis of risk. The loosely coupled relationships between the platform vendor and a complementor represent a hybrid between characteristics of a market and an alliance. Therefore, significantly new uncertainties evolve for the participants of platform ecosystems. In particular, the distribution of control and knowledge among heterogeneous actors accelerates uncertainty regarding the technology itself or the behavior of the alter. For instance, the platform vendor´s control over boundary resources (i.e. software development kit (SDK) application programming interfaces (APIs)) makes complementors increasingly dependent [6]. This limits third-party developers' space to control the exchange with the platform vendor itself. Furthermore, as this new organizing logic of digital innovation frequently requires coopetition (i.e. simultaneous cooperation and competition) to drive innovation, complementors may suffer from platform vendors to adopt and modify their applications in order to capture attractive market niches. While platform vendors encourage the development of third-party innovations, the loss of intellectual property is therefore a common threat in this context.

Such risks might result in several unfavorable results for both each complementor and the total ecosystems. Consequences of risk can be divided into direct and indirect effects and might affect not only a single complementor but also the whole platform and its ecosystem. First direct effects result directly from specific risk factors like for instance, the imitation of complementors´ products, the raise of partners to competitors or the loss of attractiveness of the platform, a delay of complementors´projects, increased costs, or an image loss. However, these first order effects can lead to indirect effects for complementors as well as the platform. Such effects include a loss of traction of the platform, platform desertion, a low innovation output of a single complementor or even the whole ecosystem. Furthermore, if the importance of the platform ecosystem is high for the complementor, a high level of risks can result in low complementor performance.

The purpose of this research is, thus, to provide insights on such risks, the drivers and relation specific nature as well as governance and architecture mechanisms to manage risks.

3. The case studies

To exploratory and descriptive single case study research design that particularly allows for researching unexplored topics. To ensure generalizable results we combined complementor cases of various cloud platforms within the context of enterprise application software and the internet of things. For an in-depth examination, multiple data sources were utilized, including a total of 21 formal interviews, internal company documents, academic publications, media sources in an exploratory study and 42 questionnaire responses on governance mechanisms from complementors of five leading cloud platforms (i.e. Microsoft Azure, Oracle Cloud Platform, Amazon Web Services, SAP HANA, and Salesforce Force.com) in study 2 [13].

Industry	Platforms	Complementors
Enterprise Application Software	3	3
Internet of Things Applications	4	5
Cloud Communications	1	1
Healthcare	1	1
Internet Videos	1	2

Platform Vendor	Platform	Complementors
SAP	HANA	9
Microsoft	Azure	9
Amazon	AWS	6
Oracle	Oracle Cloud	4
Salesforce	Force.com	14

Figure 1 Description on Cases

4. Software Ecosystem Risks

Risk factors for complementors are conditions, which can pose a serious threat to the successful development of complementary applications. Our exploratory investigations reveal a comprehensive collection of single risk factors, which can be categorized into four dimensions: market risk, technological risk, contractual risk and technological risk.

Figure 2 Risks of Software Ecosystems

Such risks emerge along the complex interactions within software ecosystems and can be assigned to single relationships and entities. Therefore, interaction models are most suitable to visualize and structure risks. The interaction model includes the partners and relationships of the complementor with its ecosystem and provides a suitable frame for organizing our findings. A high-level interaction model with its entities and relations among them is shown in Figure 2. It shows service and payment relationships between a platform vendor, the complementor and the customer. The platform vendor provides software to be resold to the complementor, who resells this software to the customer (Figure 3). On a detailed level, there are many, more specific relationships between platform vendor and complementor, which are shown in Figure 4.

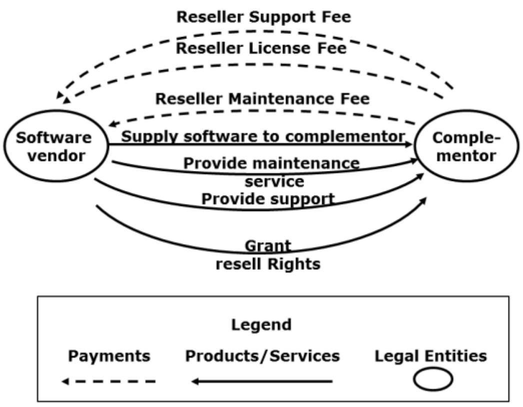

Figure 3 Interaction Model of Complementor and Platform Vendor

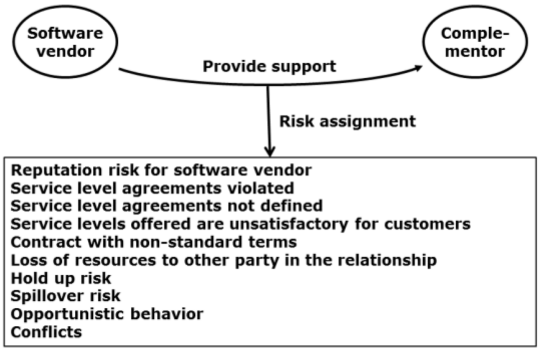

Figure 4 Risks on Single Interaction Level

5. Sources of Risk

Several sources create such risk factors. After assigning each risk to a certain relation, it is crucial to understand the sources that create and reinforce risks. This enables the complementor to start interventions against such risks.

The first source of uncertainty is ***dependency***. Such dependencies can arise from the platform on a technological base, related to specific tasks or resources from the platform vendor or ecosystem partners. Platform specificity refers to the transferability of a complementors´application to a different platform] as well as the value of complementors´assets within alternative partner relations. For instance, platforms require investments in relation-specific knowledge to participate in the platform ecosystem and capitalize from the ecosystems assets. Specific assets can be for instance, human assets, technological assets or knowledge about platform architecture, interface specifications and market characteristics. The need to get access to APIs and SDKs, the required interoperability and information about technological specifications, or the access to marketplaces like app-stores make complementors increasingly dependent. Moreover, the prerequisite investments of a complementor to participate on a platform might increase the amount of a potential loss due to a lock-in.

Second, ***market uncertainty*** constitutes a crucial driver for the risk of complementors, as for instance the sustainability of the specific niche is required to succeed. Volatile customer demand, the unpredictable emergence of new substitute products or changes in the competitive environment might increase the threat of failure during the development of complementary products.

Third, ***technological uncertainty*** covers the inability to accurately forecast the technological requirements within the relationship, which is especially important in complementary platform markets. Technological complexity and changes are the most significant sources of uncertainty. Technological uncertainty is also frequently related to a lack of experience with the technologies employed in the ecosystem, which increases the threat of failure due to inadequate capabilities. Furthermore, the unpredictability of technological evolution might constitute a source of risk during third-party innovation.

Fourth, the ***behavior of other actors in the ecosystem*** including the platform vendor itself constitutes a crucial driver for risk. When partners within the ecosystem follow their individual interest and behave opportunistically, they can might create a sense of unfairness, reduce trust and therefore increase the probability of complementors´risks. Such opportunistic behavior causes hidden costs by inefficient and ineffective processes. Moreover, although platform vendors encourage the development of complementary products to nurture the overall value of the ecosystem, there is often a tension between them and complementors. This tension arises from the complementors´threat of opportunistic behavior of the platform vendor by for instance exploiting resources or competing in the partner's niche. This driver underlines the social aspects of software ecosystems.

Finally, ***complexity*** drives the probability and amount of risk. The technological complexity of a platform, like for example the number of different APIs or programming languages, drives the amount of risk. Furthermore, the complexity of the relation itself might increase risk when for instance a partner has a lack of competence to perform certain tasks or if the complexity of third-party innovation exceeds the capabilities of a complementor. Factors like for instance the huge range of solutions, the difference in size of companies within the ecosystem as well as the multiple types of partnerships (e.g. technology-, solution-, channel-, distribution-, integration partners, and consultancies) drive complementors´ risk.

6. Managing Risks in Software Ecosystems

Suitable governance mechanisms are particularly relevant to manage the complementors´ risk in software ecosystems. While control mechanisms are mainly introduced by the platform vendor, complementors can apply architectural governance mechanisms to reduce hazards.

Control Mechanisms

One central control to govern partners and the interaction within an ecosystem is by releasing norms, mutual values and goals that are beneficial for the platform. When either a complementor or the platform vendor shows deviant behavior, other members of the ecosystem might react with social sanctioning. As a result, complementors can increase trust on platform vendors.

A suitable formal control mechanism for platform ecosystems is input control. It describes the degree to which platform vendors control complementary apps by utilizing application and selection processes. Hence, not all complementary apps are admitted to the ecosystem. Input control keeps tabs on the admission to the ecosystem and allows the platform vendor to guarantee interoperability, quality or the fit with the platform's interests, values, and positioning.

Apart from different modes of control, another central element of platform governance is the degree to which decision rights are centralized or delegated. This form of governance encompasses different classes of decision rights about what an app should do (e.g., features and functionality), how it should do it (e.g., design, user interface), and the control of boundary resources (e.g. the platform´s interfaces) among itself and the complementors. Though platform vendors are often willing to delegate decision rights to complementors because these possess nuanced knowledge about the app's means and ends, in the case of strategically relevant extensions, owners may decide to keep or retract that authority.

While platform vendors might benefit from governance mechanisms which grant them power and authority over the development of the ecosystem, these mechanisms are quite likely to confront the platforms complementors with several hazards. Compared to the centralization of decision rights, input control seems to be a particularly important mechanism. If input control is applied, all types of hazards are likely to be high. However, if input control is absent, risks are reduced. The screening and admission procedures of the platform vendor consequently require specific investments to meet such criteria. Furthermore, such control fosters uncertainty as it gives the platform vendor a certain amount of power. Input control therefore represents an essential parameter which platform vendors should calibrate carefully in order to balance own as well as complementors' risk and thus ensure healthiness and robustness of the ecosystem.

Second, the results indicate that mutual values and norms may be an effective mechanism for the platform vendor to lower complementors' risks. Such norms, mutual values and goals as soft power instrument help to lower uncertainty. Hence, clan control might to a certain degree be a suitable measure to lower the negative effects of input control and decision rights centralization.

Architectural Governance

While the control mechanisms represent a design choice on the behalf of the platform vendor, the individual complementors may choose corresponding risk mitigating governance mechanisms on their own by applying architectural governance mechanisms. For instance, the level of app decoupling describes an architecture in which changes within the architecture of the platform do not have any ripple effect on the single app. The more decoupled an app is, the more independently it can be developed by a complementor while still ensuring fluent interoperation with the platform. Usually, the complementor makes such a design choice within the exogenous constraints of the platform and minimizes the platform dependencies on the minimal degree to which an app is required to be conforming to the specifications interface This is achieved by carefully selecting and placing "thin connections" between app and platform while removing the remaining ones so that changes to the app or the platform do not condition changes to the respective counterpart. On the other hand, the standardization of interfaces describes the degree to which the linkages between the single app and the platform are stable, formalized and well-documented. Thereby, stability is ensured by the existence of boundary resources like application programming interfaces (APIs). Such standards codify the relationships between the app and the platform as well as clearly articulate rules and specifications for apps and platform infrastructure. Such clarity and transparency might help to overcome issues of opportunism and bounded rationality, so that transaction costs can be reduced. Our findings reveal, that such application architecture represents not a direct control mechanism to govern the platform dependencies of complementors. Standardization of interfaces rather represent a necessary condition to achieve a low level of risk under certain circumstances. Consequently, the use of standardized interfaces is required to minimize risk. However, if apps are highly modularized, this does not necessarily imply low levels of risk, but the effect rather depends on the environment.

References

[1] Cusumano, M. (2010), "The evolution of platform thinking", In: Communications of the ACM, 53(1), 2010, pp. 32-34.

[2] Benlian, A.; Hilkert, D. & Hess, T (2015), "How open is this platform? The meaning and measurement of platform openness from the complementors' perspective", In: Journal of Information Technology 30(3), 2015, pp. 209-228.

[3] Tiwana, A. (2015), Evolutionary Competition in Platform Ecosystems, In: Information Systems Research 26(2), 2015, pp. 266-281.

[4] Yoo, Y., O. Henfridsson, and K. Lyytinen (2010), "Research commentary-The new organizing logic of digital innovation: An agenda for information systems research", Information Systems Research, 21(4), 2010, pp. 724–735.

[5] De, B. (2017), API Management, Berkeley, 2017.

[6] Ghazawneh, A. and O. Henfridsson (2013), "Balancing platform control and external contribution in third-party development: the boundary resources model", In: Information Systems Journal, 23(2), 2013, pp. 173–192.

[7] Gawer, A. (2014), "Bridging differing perspectives on technological platforms – Toward an integrative framework", In: Research Policy 43(7), 2014, pp. 1239-1249.

[8] Jansen, S., Brinkkemper, S., and Anthony F. (2009), "Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems." In: IWSECO@ ICSR, 2009.

[9] Kude, T.; Dibbern, J. & Heinzl, A. (2012), "Why Do Complementors Participate? An Analysis of Partnership Networks in the Enterprise Software Industry", In: IEEE Transactions on Engineering Management 59, 2012, pp. 250-265.

[10] Adner, R. und Kapoor, R. (2010), "Value creation in innovation ecosystems - How the structure of technological interdependence affects firm performance in new technology generations", In: Strategic Management Journal 31(3), 2010, pp. 306-333.

[11] Tiwana, A. (2015), "Platform Desertion by App Developers.", In: Journal of Management Information Systems 32(4), 2015, pp. 40-77.

[12] Barki, H., S. Rivard, and J. Talbot (1993), "Toward an assessment of software development risk", In: Journal of Management Information Systems 10(2), 1993, pp. 203–225.

[13] Venkatesh, V., Brown, S. and Bala, H. (2013), "Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems.", In: MIS quarterly 37(1), 2013, pp. 21-54.