

# Challenges and Strategies for Managing Requirements Selection in Software Ecosystems

Daniela Damian  
University of Victoria  
Canada  
danielad@uvic.ca

Johan Linaker  
Lund University  
Sweden  
johan.linaker@cs.lth.se

David Johnson  
University of Victoria  
Canada  
davidjo@uvic.ca

Tony Clear  
Auckland University of Technology  
New Zealand  
tony.clear@aut.ac.nz

Kelly Blincoe  
University of Auckland  
New Zealand  
k.blincoe@auckland.ac.nz

## 1 Introduction

The last decade, deemed the age of the platform [1], saw a major shift in how software organizations operate and leverage platforms as a flavor of open innovation to extend their markets or “grow the pie” [2]. These platforms are used to underpin and form Software Ecosystems (SECOs) through which the platform provider, also known as the *keystone organization*, can partner and innovate together with other organizations [3]. Examples include the popular tool suite Atlassian<sup>1</sup> or the communication hub Slack<sup>2</sup>, which maintain thriving Marketplaces comprised of add-ons or third-party software integrations that extend the functionality of their own offerings.

In a SECO (see Box 1 for definition of terms), the keystone has to consider the wishes and needs not just from its own end-users, but also its ecosystem. More specifically, there are three different but sometimes overlapping sets of stakeholders: the *end-users* of its own offering, the *complementors* to the platform (i.e. the third-party organizations offering solutions and services), and the *complementors’ end-users*. Success no longer depends solely on the keystone’s efforts to manage the expectations of end-users of its own offering, instead it shifts to managing a much more complex set of business relationships within the

---

<sup>1</sup><https://www.atlassian.com>

<sup>2</sup><https://www.slack.com>

SECO [3]. The situation becomes even more complex when these relationships include collaboration between competitors, i.e., a state of *co-opetition* [4].

In this context, *requirements selection*, i.e., deciding which requirements to implement in the software platform [5] to enable extended functionality provided by the surrounding ecosystem of complementors, is a pivotal practice in order to stay ahead of the market and gain a competitive advantage. Although the ecosystem offers unprecedented opportunities for the keystone and its complementors to co-innovate in addressing market-needs [4] while also sharing the risk of betting on the wrong requirements, it is not without friction.

Managing the various requirements within the ecosystem brings challenges to the partnerships among the keystone organization and the complementors. For example, being too greedy and not leaving enough use cases for the ecosystem to innovate on or create new niches from, may hurt the ecosystem health and cause complementors to leave, and thereby have a negative impact on the keystone itself [6]. Also, leaving out requirements with a high level of innovation potential may have a negative impact on the keystone’s own competitive edge [7].

Due to these challenges and the limited focus received by existing research [8], we analyze the requirements selection processes that support this co-innovation approach in a SECO. From an in-depth study of two successful platform providers and their respective SECOS, Xero<sup>3</sup> and Shopify<sup>4</sup>, we explore and contextualize challenges related to the scaling of requirements selection in large SECOS. We interviewed members of the Ecosystem Leadership teams both at Xero and Shopify. Details about these two ecosystems and the roles we interviewed are provided in “Box 2: The Ecosystems We Studied: Xero and Shopify”. We also analyzed secondary data found in interviews in a series of podcasts with ecosystem managers and technical leaders from other established platforms: Slack, Salesforce, and Hubspot. Slack<sup>5</sup> is a communication platform; Salesforce<sup>6</sup> provides a customer-relationship management service and also sells a complementary suite of enterprise applications focused on customer service, marketing automation, analytics, and application development. Hubspot<sup>7</sup> is a platform of marketing, sales, customer service, and customer-relationships management software.

As a way for actionable insight, we also describe how Xero and Shopify manage requirements selection, and provide a set of guidelines for other organizations considering developing software within an ecosystem of partnerships with other organizations.

---

<sup>3</sup>[www.xero.com](http://www.xero.com)

<sup>4</sup>[www.shopify.com](http://www.shopify.com)

<sup>5</sup>[slack.com](http://slack.com)

<sup>6</sup>[salesforce.com](http://salesforce.com)

<sup>7</sup>[hubspot.com](http://hubspot.com)

### Box 1. Background: Software Ecosystems and Open Innovation

A Software Ecosystem (SECO) consists of a set of actors united under a common vision and aiming to solve a common problem, often through the help of an underpinning technological platform. The actors collaborate, and potentially also compete in a shared market for software and services [3]. There are many examples of successful SECOs with underpinning platforms, both open [4] and proprietary [9]. Examples include operating systems (e.g., Microsoft's Windows and Google's Android), web browsers (e.g., Google's Chrome and Mozilla Firefox), and smart home assistants (e.g., Amazon's Alexa and Apple's Siri) [9]. To provide access to their underpinning technology and enable complementary services, keystone organizations typically provide access to an open Application Programming Interface (API). Open APIs allow organizations to share functionality, while allowing their core technologies to remain proprietary, fostering open innovation within their ecosystems.

Open Innovation (OI) is an emerging field of research that aims to better understand how organizations "purposively manage knowledge flows across organizational boundaries" for improved organizational innovation [2]. Chesbrough and Bogers [2] describe three knowledge flows, modeled in Figure 1:

1. *outside-in*, where knowledge flows from external sources to improve internal innovation processes,
2. *inside-out*, where internal knowledge flows outside the organizational boundaries to external entities for innovation, and
3. *coupled*, where knowledge flows bidirectionally between the innovating actors.

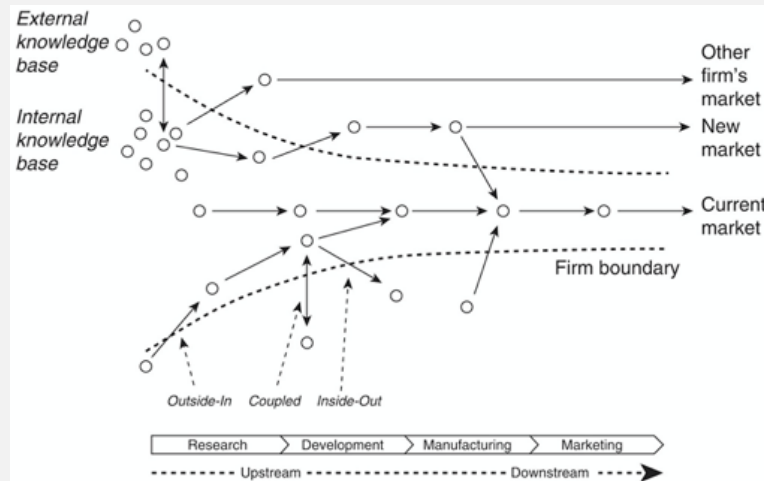


Figure 1: The open innovation model by Chesbrough and Bogers[2] where the inside of the funnel represents the inside of the company, and the funnels' borders represent the company's wall to the outside through which the different knowledge flows (outside-in, inside-out and coupled).

Most OI research has focused on outside-in innovation, while inside-out and coupled OI are less understood in the literature [2]<sup>4</sup>

Within a SECO, a keystone organization engages in partnerships and open innovation that may involve all three forms of knowledge flows among the innovating SECO actors, described in detail in Section 2: innovation to the platform can be leveraged by external partners who address new use-cases, resulting in markets grown and penetrated [3].

**Box 2. The Ecosystems We Studied: Xero and Shopify**

**Xero** is a software producing organization that provides accounting software-as-a-service to small businesses and their advisors. One of the keys to Xero's success is that in addition to producing accounting software, they also provide access to a service platform that enables other software developers to access Xero data through the platform. This has enabled a SECO of complementors that extend and enhance the Xero product for added value to the users of the Xero SECO. Offering this base platform, building relationships with its complementors through a tiered partnership program, and hosting a shared marketplace has allowed Xero to evolve from a company that primarily offered an accounting solution to a SECO that enables a full suite of small business solutions with its own offering at the core. As of March 2020, Xero had more than 750 partners in its Marketplace. We collected data on the Xero SECO through interviews with the Xero Executive Ecosystem Leadership team such as the Executive Ecosystem GM, Developer Experience GM, Ecosystem Partnership GM, Customer Experience GM, and Ecosystem Marketplace GM, as well as the CEO's of several of the ecosystem partner organizations (anonymous, to protect their confidentiality).

**Shopify**, the second organization we investigated, is also taking advantage of service platforms to help grow their business. The core Shopify product provides merchants with a set of tools to simplify the creation of eCommerce websites. Similar to Xero, Shopify has opened up their eCommerce technology and data to third-party developers through a service platform. Using the service platform, complementors are able to create new extensions and plugins to meet niche needs of merchants not met by the core Shopify eCommerce product. Opening internal data and functionality through the service platform has enabled Shopify to create a rich SECO that provides extended value to Shopify merchants. As of March 2020, Shopify had more than 2500 partners in its Marketplace. Similar to Xero, we interviewed and analyzed input from Shopify's Platform Development Director and Chief Technology Officer.

Our analysis of these interviews at Xero and Shopify used thematic analysis, whereby we searched for emerging themes with respect to how requirements engineering was carried out within the ecosystem, challenges encountered, as well as strategies that the keystone organization employed in their requirements selection within the SECO. The challenges we report here predominated and were largely similar across the two ecosystems. Quotes from each ecosystem indicate specific situations at each ecosystem. The insights we gained from our secondary data corroborated these challenges; similarly, the strategies and overall approach were similar across the three additional ecosystems analyzed.

### Box 2 (cont'd). Horizontal and Vertical Integrations in the Xero and Shopify ecosystems

The complementary software and services enabled by the provisioning of the open Application Programming Interfaces (APIs) in these ecosystems are provided in third-party apps and typically referred to as *vertical* or *horizontal integrations* onto the ecosystem platform. The *vertical integrations* focus on niche industries by customizing product features and potentially creating new markets by implementing requirements specific to users in a particular industry. For example, while Xero provides general accounting software for small businesses, any interested SECO actor may contribute their domain-specific expertise to create new value for the market (i.e. specialized functionality) or create a new market. Figure<sup>a</sup>, a complementor in Xero's SECO, for example, has created a new market by taking the platform and building a set of technologies on top of the core accounting product to enable a small business solution for farmers. Similarly, Shopify provides a core solution meeting the general needs of the wide ranging eCommerce industry, but complementors develop new apps on top of the core functionality that create, or enable, niche industries. For example, in recent years the subscription box industry has undergone rapid growth within the eCommerce market<sup>b</sup>. However, Shopify does not provide core features supporting this industry. Instead, Shopify relies on complementors to build apps to enable this market, such as Subscriptions by Recharge<sup>c</sup>. By enabling vertical solutions through the service platform, a SECO can reach markets that the keystone did not plan or know to include through the core application, leaving the requirements elicitation up to the complementors.

The *horizontal integrations* bring new functionality to the entire SECO; they focus on the breadth of industries supported by the SECO and extend value propositions and the market for the entire platform. For example, Xero provides basic reporting functionality, but *Spotlight Reporting*<sup>d</sup> (a complementor) provides extended reporting and forecasting capabilities that provide potential value to the breadth of the market, as any users of Xero and the vertical solutions can choose to integrate with such horizontal solutions. In the case of Shopify, horizontal solutions, such as AfterShip<sup>e</sup>, which provides an enhanced package tracking solution, offer general eCommerce functionality that Shopify either overlooked or chose not to implement in their core functionality. Horizontal solutions increase the value of the SECO by allowing external actors to innovate by applying their knowledge to address emergent requirements not elicited by the keystone or those that do not align with the keystone's roadmap or available resources at particular times.

Unique to software development within an ecosystem, the innovation emerging from the combination of both vertical and horizontal solutions results in a dynamic offering whereby the customers are provided with additional functionality that was not previously envisioned but that emerges. This dynamic offering is possible through the network effects in the ecosystem. For example, in using a vertical solution, Xero customers get access to a set of functionality customized to their domain, which makes the core product more accessible to niche industries. By using additional integrations (e.g. through more generic, horizontal solutions), their accounting software accesses additional functionality that exceeds their previously perceived needs but that are found to be useful.

<sup>a</sup>[apps.xero.com/ca/search/app/figured](https://apps.xero.com/ca/search/app/figured)

<sup>b</sup>[get.fuelbymckinsey.com/article/sizing-up-the-subscription-e-commerce-market/](https://get.fuelbymckinsey.com/article/sizing-up-the-subscription-e-commerce-market/)

<sup>c</sup>[apps.shopify.com/subscription-payments](https://apps.shopify.com/subscription-payments)

<sup>d</sup>[apps.xero.com/ca/app/spotlight-reporting](https://apps.xero.com/ca/app/spotlight-reporting)

<sup>e</sup>[apps.shopify.com/aftership](https://apps.shopify.com/aftership)

## 2 Requirements Knowledge flows and Co-innovation in a SECO

A SECO adds complexity to the RE process for a software producing organization turned service platform provider. However, operating within the SECO, the keystone organization has the unique opportunity to leverage open innovation through synergistic relationships with the complementors in the SECO and their markets. The different complementors provide sources and venues for requirements elicitation.

Once elicited, the keystone will then select the requirements they themselves are to implement and invest in, and which ones to leave out for the complementors. The requirements selection within the SECO is therefore part of a *co-innovation* process whereby the requirements selected by the keystone guide the innovation on the platform, in turn enabling the complementors to build on this new functionality and address the SECO requirements not tackled by the keystone. Some of the new functionality offered by the complementors will generate emergent requirements that feed back into the keystone; this enables the keystone to further innovate to increase the platform offerings, creating a circular requirement knowledge flow. We use the terms use cases, features and requirements knowledge interchangeably as representation of the information that becomes available in this circular flow, and that contributes directly to requirements in the ecosystem.

Figure 2 describes the elements of this requirements knowledge flow within a platform SECO, illustrating how the keystone organization leverages open innovation to grow its platform in a way that mutually benefits its partnerships within the SECO. Relationships between the external and internal knowledge bases in the SECO, in the forms of *outside-in*, *inside-out* and *coupled knowledge* [2] are explained below, together with sources of such knowledge, as well as venues through which this knowledge is communicated or elicited.



knowledge of the platform, they interact directly with app developers in explaining and providing ongoing API support. At the same time they are a key source of feedback to the platform on API usage and challenges faced by app developers, therefore informing API requirements and API info necessary to maintain productive relationships with the app developers.

In the ecosystem, *complementors* innovate on emerging market trends that complement the features of the keystone’s core product. The complementors have their own channels for requirement elicitation; *feature requests from end-users* signal user requirements, allowing complementors to leverage domain-dependent, niche use cases (in vertical solutions) or generic use cases (in horizontal solutions) that the keystone has not implemented, and in turn inform new opportunities for platform growth. Not only do the new use cases inform the requirements for complementors’ services, but they also provide important knowledge that drive new API requirements, which in turn foster innovation in the complementors.

Further, understanding the potential value from new functionality made available on the platform, therefore, becomes paramount for the App developers. App developers learn about the current or planned *API information*, and the opportunities enabled by the APIs, through Developer conferences organized by the keystone organization or through conversations with developer evangelists. The keystone also typically shares a *platform roadmap*, available on the keystone website and also often communicated through developer evangelists. The roadmap aims to inform complementors of upcoming features and API information to help strategically guide innovation.

*“This is something that is great in theory and harder in practice. .. we don’t tell developers exactly what to develop, but [ ..] in annual developer conferences we think deeper on what we are going to present so that we tell them where the opportunities are (more like thematic than specific), and here are the new APIs that are available to build into”* (Platform Development Director, Shopify).

Finally, Developer conferences are also venues where the keystone organization signals to complementors open areas in which they can innovate, by connecting end-users and customers with app developers so that requirements from potentially emerging use cases on the public APIs can be fleshed out.

### 3 Challenges in Requirements Selection within a SECO

Our description of the ecosystem elements and its requirements knowledge flow so far has not considered the dynamic nature of the ecosystem environment. It inherently provides a rather static snapshot of a business and software engineering environment that is complex and dynamic. As Shopify’s CTO explains, *“[the platform ecosystem] is constantly moving and there are symbiotic and parasitic relationships between actors in the ecosystem (in horizontal and vertical solutions). It’s hard to see this ecosystem as a snapshot in time, as it’s the*



*evolution over time that shows which actors have had good or bad strategies.”*

Selecting requirements within a dynamic SECO relates to managing business partnerships and watching over the ecosystem’s health as much as considering decisions of a technical nature. From our data from Xero and Shopify, as well as the secondary data from the three other ecosystems, we distilled a number of challenges the keystones face in their requirements selection within the dynamic ecosystem.

*Challenge 1. The platform provider’s decisions around which of the elicited requirements they will implement and which ones should be left for the complementors is a balancing act.* By leaving requirements for the complementors and encouraging innovation within the SECO, platform providers can monitor their SECO marketplace to determine which new products are adding the most value to the market and make strategic decisions based on this, e.g., acquisition or extended support of a complementor. While implementing requirements in the core product may improve their product offering technically, it may reduce the opportunity for complementors to join and grow within the SECO. This risks the SECO’s health as it can damage relationships built over time and the overall value proposition of the platform provider [6].

*Challenge 2. Managing feedback from third-party app developers as key end-users is difficult and requires patience and diplomacy.* First, app developers, knowledgeable in both technical aspects of the platform and end-user needs, can be harsh and demanding of functionality that advances specific applications. Yet, they need to be handled with respect because they are a key link in the adoption of platform features. Second, the feedback on new platform features is inevitably delayed since the third-party app developers need time to implement new functionality which consumes the newly available APIs and release that functionality to their end users. These delays make it difficult to quickly assess the value of new platform features.

*Challenge 3. Growing the core keystone product functionality through acquisitions from the SECO while ensuring customer adoption.* Often, the keystone’s response to successful (often horizontal) solutions is to acquire a particular complementor, and, thus, engage in a situation of *co-opetition* with other complementors in the marketplace offering similar functionality. This poses a challenge in attracting customers to use the newly acquired functionality, while ensuring the health of the partnership with the competitors in the SECO.

*Challenge 4. Managing undesired latent requirements while maintaining control within the SECO.* When opening up the platform, not all functionality created by the complementors is envisioned by the keystone, since at times requirements can be elusive, subsuming aspects that can be recognized but not defined. Some new functionality that emerges through innovation in the complementors might turn out to be very useful for the customers – latent or unexpected requirements. Yet, the keystone might want to keep control over such functionality for reasons such as revenue growth and data security, particularly for features that involve confidential customer data.

## 4 Requirements Selection through Incremental Investment and Risk Sharing

In this section, we describe an approach to manage requirements selection in a SECO. Similar to the challenges above, we distilled this approach from our study participants and how they address the overarching challenges above. There is not a one to one mapping between challenges and strategies, as some guidelines might address multiple challenges. While these strategies may appear related to business decisions, as is apparent in Jansen’s model of SECO governance [3], the boundaries between software engineering and business decisions are now increasingly blurred, so these are important to the software development process as a whole. We also outline a number of guidelines for practitioners; they are intended to be fairly generic, their actual implementation will depend on the particular organization and their context.

### 4.1 The SECO 80/20 rule in selecting requirements

Platform and SECO managers described following the Pareto principle, or the 80/20 rule, when devising their product and platform strategies for deciding which requirements to pursue and which to leave to the SECO. By keeping an eye on the current market needs, the keystone exercises *adaptive capabilities* [10] to identify and take advantage of resulting opportunities. This response is typically pragmatic and opportunistic based on local market circumstances and user preferences.

#### Balancing across the ecosystem

The keystone’s general strategy is to develop core functionality that meets the needs of roughly 80% of the market, leaving the complementors to fill the remaining 20%. The 80% represent the use cases that have widest reach within the market. The keystone opens APIs to enable the complementors to fill the remaining 20%. Figure 2 shows how the core product typically satisfies 80% of use cases, while the platform enables the complementors to satisfy the remaining 20%. The keystone needs to balance the ecosystem’s health and the complementors ability to innovate and create new niches, against the keystone’s business and ecosystem strategy [7]:

*“[it] comes back to ‘what do we want to be the first mover on?’ what are we comfortable in letting other people do or own?”* (Developer Experience General Manager, Xero)

*“Decisions would be based on [...] a combination of monitoring, constant assessment and constant re-evaluation of what’s going on (and that is the more controlled side) and then the willingness to roll the dice”* (Platform Development Director, Shopify)

To grow and create a healthy SECO requires governance from the keystone [11] whereby keystones exercise their power wisely [12] and are careful to not outcompete nor limit the innovation potential for the complementors, while still capturing enough value themselves to stay profitable and be able

to reinvest into the SECO and its underpinning platform [9]. Two factors to consider in these decisions are 1) a feature’s position in the commoditization cycle (e.g., *innovative* to *differentiating* to *commodity* product) and 2) the general stance of the platform provider’s product compared to competition on the market [7].

*Guideline 1: Leave the complementors within the SECO to identify and manage the 20% of use cases, as part of enabling innovation within the ecosystem.*

### Enabling the ecosystem

As previously described, the creation of the platform with public APIs is the key enabler to understanding requirements that can be leveraged by the ecosystem. The open APIs enable both the keystone and its complementors to *share the risk* of betting on the wrong requirements [3]. By leaving certain requirements or opportunities for the complementors, the keystone can see how they play out [13]. If the functionality proves successful, the keystone is then in a position to integrate the functionality into the platform or product, acquire the complementor, or partner with them. If the functionality did not play out well, the keystone has avoided the consequences of taking the risk at the expense of the complementor.

This aligns with the real options theory [13] of limited and incremental investment as well as staged commitments based on reducing uncertainty through increased knowledge to manage risk. The keystone first invests in the development and documentation of APIs, then provides ongoing support for these APIs to app developers (through developer evangelists). This creates options for the keystone for later investments in the form of buying, building, or partnering as new functionality becomes successful (described next).

*Guideline 2: Provide open APIs with rich documentation along with development support and evangelism to enable complementors of the ecosystem to explore the 20% of use cases left out in the ecosystem.*

### Maintaining the ecosystem

Thus, managing the third-party app developers who use this platform then becomes a core capability of the keystone. Coping with sometimes brutal feedback of the complementors as demanding and technically knowledgeable platform users is a necessary evil<sup>8</sup> in developing API requirements and ensuring that the APIs are successfully leveraged within the ecosystem to enable innovation.

*Guideline 3: Embrace the developers’ feedback from complementors within the SECO and support them in their innovation process.*

## 4.2 Innovation through Build/Buy/Partner

Having identified the opportunities provided by the emerging requirements, the keystone is in an enviable, unique position to exercise their *innovative capability*.

<sup>8</sup>[devblog.xero.com/developers-as-your-customer-or-how-i-learned-to-stop-worrying-and-love-the-feedback-loop-19a3caa62aa3](https://devblog.xero.com/developers-as-your-customer-or-how-i-learned-to-stop-worrying-and-love-the-feedback-loop-19a3caa62aa3)

ity [10].

### **Build/Buy/Partner decision**

The keystone can decide to *build* new functionality in-house, *buy* (or acquire) one of the complementors that has proven successful implementation of the requirements, or *partner* with the respective complementor [14].

*“Whenever we look at a new area or geography, we are constantly asking ourselves what are we uniquely positioned to build, what are uniquely position to build but don’t have that capacity or know how and then we go into an acquire situation or, in two cases, either we are not sure what it is so let’s rev up the APIs and see what people do with them, or partner”* (Platform Development Director, Shopify)

*“it is part of your product strategy, ... which is these are the segments of the customers that you want to own, and these are the main needs that they have, in each market, then for your product and each value proposition it has to deliver, you decide whether you want to build it, to partner, or to buy”* (Ecosystem Executive Manager, Xero)

The keystone then monitors the complementors’ offerings. Should the complementors be seen to develop any strategically valuable functionality, the keystone will then make the build/buy/partner decisions. This process is enabled by tiered *partnership* programs that enable the keystone to collaborate with complementors through various levels of shared roadmapping and provide support for the co-innovation. *Acquiring* a complementor (most often for horizontal solutions) is a response that enables the keystone to bring additional functionality into the core platform.

<i>Guideline 4: Continually monitor the complementors within the SECO to identify opportunities to build, buy, or partner as new use cases emerge.</i>
--

### **Strategic Partnerships**

In addition to the build/buy/partner decisions, the keystone also often develops strategic partnerships with some of the SECO actors. These partnerships bring significantly higher business value and unlock new opportunities for the keystone. For these strategic partners, requirements negotiations often occur through a tighter alignment between the roadmaps of the keystone and the strategic complementor (similar relationships can also be expected between complementors directly [4]). This tighter alignment can benefit both the keystone and the strategic partners by ensuring their roadmaps are complementary. However, the existence of strategic partners can cause problems in the ecosystem if they are not carefully managed as other complementors may feel the strategic partners have been given an unfair advantage. The keystones need to maintain a general level of trust among the complementors, as is important for SECOs in general [4].

<i>Guideline 5: Strategic partnerships must be carefully managed to ensure the remaining SECO actors do not feel slighted.</i>
--

### 4.3 Co-opetition as Orchestrating mutual benefit in partnerships

The build/buy/partner decisions can contest the said 80%-20% boundary in section 4.1, and it is not without friction at times.

#### Managing Acquisition Announcements

A first conflict arises when the keystone acquires a complementor and consequently, by absorbing its added functionality within the keystone's core offering, competes with other existing ecosystem complementors that provide similar functionality. This creates a co-opetition relationship where the complementor's offering co-exists in the SECO marketplace in competition with the keystone's corresponding functionality. This new competition must be carefully managed, given the power held by the keystone. First, it is important to communicate these acquisition decisions to the SECO actors.

*Guideline 6: Give the new (potential) competitors within the SECO a heads up about an acquisition*

#### Enabling co-opetition

The newly acquired functionality provides the keystone with the opportunity to directly compete with complementors who may already have established market segments or to focus on strengthening its presence in complementary segments. For example, Xero's acquisition of Hubdoc, a successful horizontal solution for document management, created a tension with ReceiptBank, an app in Xero's marketplace that provides overlapping functionality. However, Xero's strategy was to communicate that they were "interested in the value we provide to our own joint customers" [15] and to be honest about its intentions of strengthening their position in Australia and New Zealand by including automatic document ingestion in the core product [15]. ReceiptBank is now in direct competition, but they maintain their strong position in the UK market, and by staying in the Xero ecosystem, they have the opportunity of going into new markets where Xero already exists, such as Australia. So while Xero and ReceiptBank may operate as competitors, their cooperation through the ecosystem enables them both to share in a strategy which 'grows the pie' for each of them.

*Guideline 7: Stimulate and enable co-opetition within the SECO as there could be mutual value in the partnership.*

#### Leveraging New Functionality

One successful strategy is to make the newly acquired functionality available and, thus, commoditized through its open APIs. Taking the customers' and the overall SECO perspective, such a move is positive as the potentially innovative and differentiating feature of the acquired complementor becomes available to the entire ecosystem. The customer is now presented with higher diversity and variety of solutions generated within the platform marketplace [3], and the keystone has the opportunity to use the competing applications as a sounding board for the quality of its own offering.

Another more selective option is to not make it available through an API, keeping the functionality closed and only available to the keystone’s customers as a differentiating service. This is a balancing act as if complementors in the SECO start to experience unfavorable behaviour perceived as abuse of power from the keystone, they will look to other alternatives for their business. Hence, it may be a good idea for the keystone to negotiate with the complementor before entering into co-opetition. In such cases, the keystone must balance which types of power it chooses to exercise, i.e., use expert and rewarding approaches rather than a more up-front coercive one [12]. For example, by offering its expertise through a partnership with the complementor to jointly develop the concerned functionality and keeping it exclusive, the keystone can use its position to steer the development of the feature while minimizing the negative business impact for the complementor.

*Guideline 8: Carefully consider the impact of commoditizing the newly acquired functionality by making it available on the API.*

### Monitoring New Functionality

If the new functionality is commoditized, another conflict arises when the functionality enabled by the ecosystem opens a negative and unforeseen area of functionality from the perspective of the keystone that could be potentially exploited by the complementor, e.g., forms of data aggregation that introduce privacy or IP risks for their clients. The keystone will need to continually watch for and monitor contested use cases and govern access to such functionality. Managing the relationship with the complementor in these cases which involves closing off part of the ecosystem can be delicate. One approach may be to monetize access to new functionality or data deemed of ‘high value’ through strategic or tiered partnerships within the ecosystem.

*Guideline 9: Continually monitor API usage for unintended use cases implemented by the complementors within the SECO, i.e. emerging innovation without negative consequences such as compromised customer privacy.*

## 5 Conclusion

In a SECO, platform development organizations engage in a requirements selection process that is challenging and intertwined with broader decisions related to managing open innovation within large scale collaboration and dynamic partner relationships. By studying the approaches of a number of established software ecosystems, we identified the challenges they face as well as described an approach comprising of their strategies in an incremental investment in innovation and risk sharing process.

Slinger & Cusumano’s classification model [16] situates our ecosystem case organisations as owners of privately owned service platforms, with an extension market for complementors to offer paid products to customers. Therefore the challenges and strategies discussed here are limited to these types of ecosystems

(as opposed to platform ecosystems such as Ubuntu or open source ecosystems like Eclipse).

Nevertheless, we hope that the guidelines we describe serve as actionable insights to other organizations designing their own approaches within a service platform ecosystem, as well as create opportunities for further research in what is becoming a predominant mode of software development: software ecosystems.

## References

- [1] P. Simon, “The age of the platform: How amazon, apple, facebook, and google have redefined business,” 2011.
- [2] H. Chesbrough and M. Bogers, *Explicating open innovation: Clarifying an emerging paradigm for understanding innovation*. Oxford: Oxford University Press, 2014, pp. 3–28.
- [3] S. Jansen, “A focus area maturity model for software ecosystem governance,” *Information and Software Technology*, vol. 118, p. 106219, 2020.
- [4] A. Nguyen-Duc, D. S. Cruzes, S. Terje, and P. Abrahamsson, “Do software firms collaborate or compete? a model of coopetition in community-initiated oss projects,” *e-Informatica Software Engineering Journal*, vol. 13, no. 1, pp. 37–62, 2019.
- [5] S. Barney, A. Aurum, and C. Wohlin, “A product management challenge: Creating software product value through requirements selection,” *Journal of Systems Architecture*, vol. 54, no. 6, pp. 576 – 593, 2008, selection of best papers from the 32nd EUROMICRO Conference on ‘Software Engineering and Advanced Applications’ (SEAA 2006).
- [6] M. Iansiti and R. Levien, *The keystone advantage: what the new dynamics of business ecosystems mean for strategy, innovation, and sustainability*. Harvard Business Press, 2004.
- [7] H. H. Olsson and J. Bosch, “Strategic ecosystem management: A multi-case study in the b2b domain,” in *Product-Focused Software Process Improvement*, P. Abrahamsson, L. Corral, M. Oivo, and B. Russo, Eds. Cham: Springer International Publishing, 2015, pp. 3–15.
- [8] V. Aparna and et.al., “A systematic mapping study on requirements engineering in software ecosystems,” *Journal of Information Technology Research (JITR)*, vol. 11, pp. 49–69, 2018.
- [9] J. Bosch, *Speed, data, and ecosystems: Excelling in a software-driven world*. CRC press, 2017.
- [10] R. Lal and T. Clear, *Enhancing product and service capability through scaling agility in a global software vendor environment*. Los Alamitos, California: IEEE, 2018, pp. 59–68.

- [11] C. Alves, J. A. P. de Oliveira, and S. Jansen, “Software ecosystems governance-a systematic literature review and research agenda.” in *ICEIS (3)*, 2017, pp. 215–226.
- [12] G. Valença, C. Alves, and S. Jansen, “A power perspective on software ecosystem partnerships,” in *International Conference on Product-Focused Software Process Improvement*. Springer, 2016, pp. 69–85.
- [13] W. Vanhaverbeke, V. Van de Vrande, and H. Chesbrough, “Understanding the advantages of open innovation practices in corporate venturing in terms of real options,” *Creativity and innovation management*, vol. 17, no. 4, pp. 251–258, 2008.
- [14] M. Borg, P. Chatzipetrou, and et al., “Selecting component sourcing options: A survey of software engineering as broader make-or-buy decisions,” *Information and Software Technology*, vol. 112, pp. 18 – 34, 2019.
- [15] 12 Feb 2020. [Online]. Available: <https://www.afr.com/companies/professional-services/accounting-software-minnow-says-it-can-prosper-in-xero-s-shadow-20200208-p53yxb>
- [16] S. Jansen and M. Cusumano, “Defining software ecosystems: A survey of software platforms and business network governance,” *Proceedings of IWSECO*, p. 41, 2012.