

Author Notes

- ✓ The type has been replaced in your figures; please proof the figures carefully for any errors.
- ✓ The last page contains pull quotes that might be used in the article, depending on design and spacing considerations. Are these pull quotes appropriate? If not, please provide several alternatives. Pull quotes should be concise and taken verbatim from article text.

<AU: Please check that the feature summary above is appropriate.>

Challenges and Strategies for Managing Requirements Selection in Software Ecosystems

Daniela Damian, University of Victoria

Tony Clear, Auckland University of Technology

Kelly Blincoe, University of Auckland

David Johnson, University of Victoria

Johan Linåker, Lund University <AU: Please check whether the affiliation for David Johnson is the University of Victoria or Bielefeld University in the byline and in the author biography.>

// In platform software ecosystems, organizations partner and innovate together. Success and innovation depend on managing complex sets of business relationships and stakeholders and using a requirements-selection process. We describe the associated challenges and strategies from the study

of two large proprietary platform ecosystems.

<AU: Please note that the abstract has been trimmed in accordance to magazine style. Please check that included details are correct/okay as given.> //

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 (OI) 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 (<https://www.atlassian.com>) or the communication hub Slack (<https://www.slack.com>), which maintain thriving marketplaces comprising add ons or third-party software integrations that extend the functionality of their own offerings.

In a SECO (see “Background: Software Ecosystems and Open Innovation” for definitions of terms), the keystone has to consider the wishes and needs from not just 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 (the third-party



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 open⁴ and proprietary.⁵ Examples include operating systems (such as Microsoft Windows and Google's Android), web browsers (for example, Google's Chrome and Mozilla Firefox), and smart home assistants (like Amazon's Alexa and Apple's Siri).⁵ To provide access to their underpinning technology and enable complementary services, keystone organizations typically provide access to an open application program-

ming interface (API). Open APIs allow organizations to share functionality, while allowing their core technologies to remain proprietary, fostering open innovation (OI) within their ecosystems.

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 describe three knowledge flows,² modeled in Figure S1: **<AU: Please note that original Figure 1 is now Figure S1 and included in this sidebar. Original Figure 2 is now Figure 1.>**

1. *outside in*, where knowledge flows from external sources to improve internal innovation processes

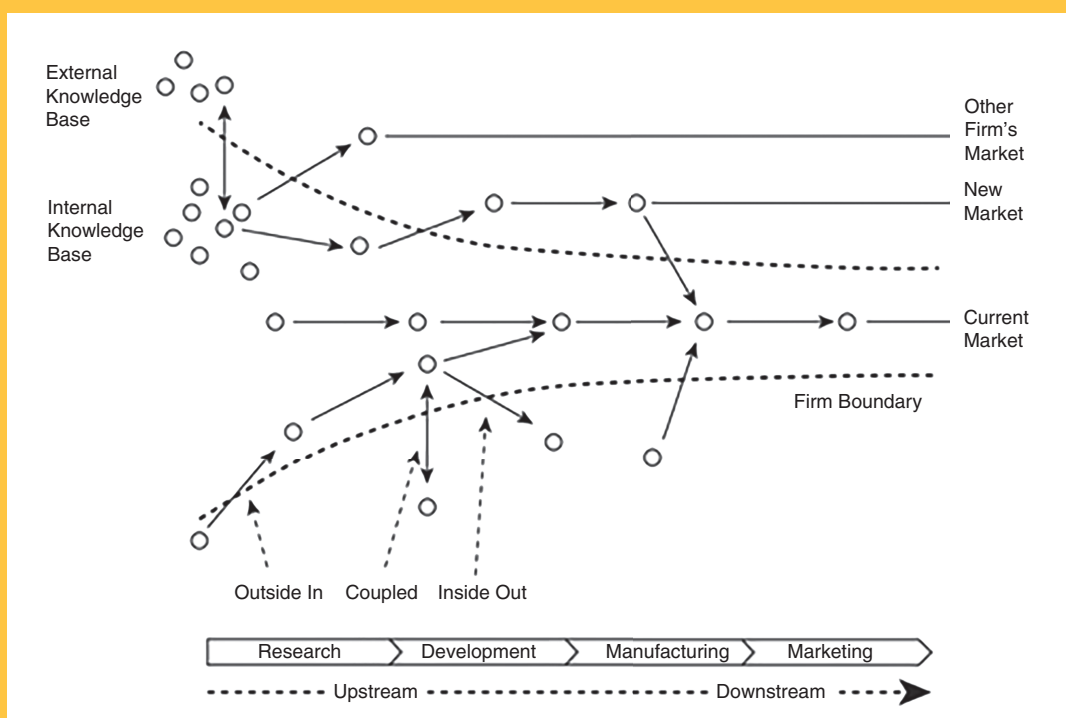


FIGURE S1. The open innovation model by Chesbrough and Bogers,² where the inside of the funnel represents the inside of the company, and the funnel's borders represent the company's wall to the outside through which the different knowledge flows (outside in, inside out, and coupled). **<AU: Please check in Figure S1 if the arrows for outside in, inside out, and coupled should all show the same direction.>**

2. *inside out*, where internal knowledge flows outside the organizational boundaries to external entities for innovation
3. *coupled*, where knowledge flows bidirectionally between the innovating actors.

Most OI research has focused on outside-in innovation; the inside-out and coupled OI flows are less understood in the literature.^{2,4} **<AU: Please check that both references 2 and 4 are correctly cited here.>**

Within a SECO, a keystone organization engages in partnerships and OI that may involve all three forms of knowledge flows among the innovating SECO actors, described in detail in the section “Requirements Knowledge Flows and Co-Innovation in a SECO.” Innovation to the platform can be leveraged by external partners who address new use cases, resulting in markets grown and penetrated.³

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 SECO.³ The situation becomes even more complex when these relationships include collaboration among competitors, that is, a state of *co-opetition*.⁴

In this context, “requirements selection,” that is, deciding which requirements to implement in the software platform⁶ **<AU: Please note that, per magazine style, in-text reference citations need to be in order. The sidebar “Background: Software Ecosystems and Open Innovation” is also involved. Please check that the references were adjusted correctly.>** to enable extended functionality provided by the surrounding ecosystem of complementors, is a pivotal practice 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 needs⁴ while also sharing the risk of betting on the wrong requirements, the situation 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 on which the ecosystem can innovate or from which it can create new niches may hurt the ecosystem health and cause complementors to leave, thereby resulting in a negative impact on the keystone itself.⁷ Also, leaving out requirements with a high level of innovation potential may have a negative impact on the keystone’s own competitive edge.⁸

Due to these challenges and the limited focus on these topics by existing research,⁹ 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 (www.xero.com) and Shopify (www.shopify.com), we explore and contextualize challenges related to the scaling of requirements selection in large SECOs. **<AU: Please note that footnotes are not permitted as per magazine style. Footnotes are incorporated into the text. Please check that the placement is okay.>** We interviewed members of the ecosystem leadership teams at both Xero and Shopify. Details about these two ecosystems and the roles we interviewed are provided in “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 (slack.com) is a communication platform; Salesforce (salesforce.com) 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 (hubspot.com) is a platform of marketing, sales, customer service, and customer-relationships management software.

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

Requirements Knowledge Flows and Co-Innovation in a SECO

A SECO adds complexity to the requirements engineering **<AU: Kindly check that the expansion of RE is correct.>** process for a software-producing organization turned service-platform provider. However, operating within the SECO, the



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, it also provides access to a service platform that allows other software developers to access Xero data. 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 General Manager (GM), Developer Experience GM, Ecosystem Partnership GM, Customer Experience GM, and Ecosystem Marketplace GM as well as the CEOs of several of the ecosystem partner organizations (we refer to them anonymously, to protect their confidentiality).

Shopify, the second organization we investigated, is also taking advantage of service platforms to help grow its business. The core Shopify product provides merchants with a set of tools to simplify the creation of e-commerce websites. Similar to Xero, Shopify has opened up its e-commerce technology and data to third-party developers through a service platform. Using this platform, complementors are able to create new extensions and plug-ins to meet niche needs of merchants not met by the core Shopify e-commerce 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 2,500 partners in its marketplace. Similar to our procedure with Xero, we interviewed and analyzed input from Shopify's Platform Development Director and CTO.

From these interviews at Xero and Shopify, we used a 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 ecosystem-specific situations. 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.

The complementary software and services enabled by the provisioning of the open application programming interfaces in these ecosystems are provided in third-party apps. They are typically referred to as *vertical* or *horizontal integrations* onto the ecosystem platform.

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 its domain-specific expertise to create new value for the market (specialized functionality) or create a new market. Figured (apps.xero.com/ca/search/app/figured), 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 e-commerce industry, but complementors develop new apps on top of the core functionality that create or enable niche industries. For instance, in recent years the subscription box industry has undergone rapid growth within the e-commerce market (get.fuelbymckinsey.com/article/sizing-up-the-subscription-e-commerce-market/). 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 (apps.shopify.com/subscription-payments). 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.

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 (a complementor; see apps.xero.com/ca/app/spotlight-reporting) 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 (apps.shopify.com/aftership), which provides an enhanced package-tracking solution, offer general e-commerce functionality that Shopify either overlooked or chose not to implement in its 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 road map 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 emerges, although it was not previously envisioned. 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 functionalities customized to their domain, which makes the core product more accessible to niche industries. By using additional integrations (for example, through more generic, horizontal solutions), their accounting software accesses additional functionalities that exceed their previously perceived needs but that are found to be useful.

keystone organization has the unique opportunity to leverage OI 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 that they themselves are to implement and invest in and those 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 representations of the information that becomes available in this circular flow and contributes directly to requirements in the ecosystem.

Figure 1 describes **<AU: Please note that original Figure 2 is now Figure 1. Original Figure 1 is now Figure S1 and included in the sidebar “Background: Software Ecosystems and Open Innovation.”>** the elements of this requirements knowledge flow within a platform SECO, illustrating how the keystone organization leverages OI 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 knowledge² are explained in the following paragraphs, together with sources of such knowledge, as well as venues through which this knowledge is communicated or elicited.

In the keystone organization, innovation lies in the capabilities of the platform to act as a bridge between app developers and the core product family, thus enabling new extensions and integrations in the form of third-party apps. Two main sources of requirements knowledge exist for the keystone organization: the end users of its own core product and the third-party apps, and the third-party app developers, who are end users of the platform itself. For the ongoing success of its SECO, the platform now requires ease of use and visibility of its public application programming interfaces (APIs) to enable its complementors to better innovate. The app developers have knowledge of both the end users and the platform APIs and therefore become an essential conduit of requirements knowledge in the innovation process.

External knowledge that informs platform and API growth translates to API requirements. This external knowledge includes 1) feature

requests from end users, 2) feedback on the quality of third-party apps from end users, and 3) feature requests from app developers. Feedback and requests from end users across the SECO are captured through the keystone's marketplace app reviews. Feature requests from app developers come through online developer community Forums, Stack Overflow, and social media (such as Twitter and LinkedIn). Feature requests from both app developers and end users are also made through discussions at developer conferences. Developer conferences connect platform developers, developer evangelists, app developers, and end users in exchanging feedback and ideas for new platform features.

The role of developer evangelists in the keystone organization is unique to a platform development environment. With strong development skills and 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 providing the API requirements and information 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. <AU: Please check that the preceding edited sentence conveys the intended meaning.> Not only do the new use cases inform the requirements for complementors' services, but they also provide important knowledge that drives new API requirements, which in turn foster innovation in the complementors.

Further, understanding the potential value from new functionality made available on the platform thus 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 road map, available on the keystone website and also often communicated through developer evangelists. The road map 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.

Challenges in Requirements Selection Within a SECO

So far, our description of the ecosystem elements and its requirements knowledge flow 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.

1. *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,*

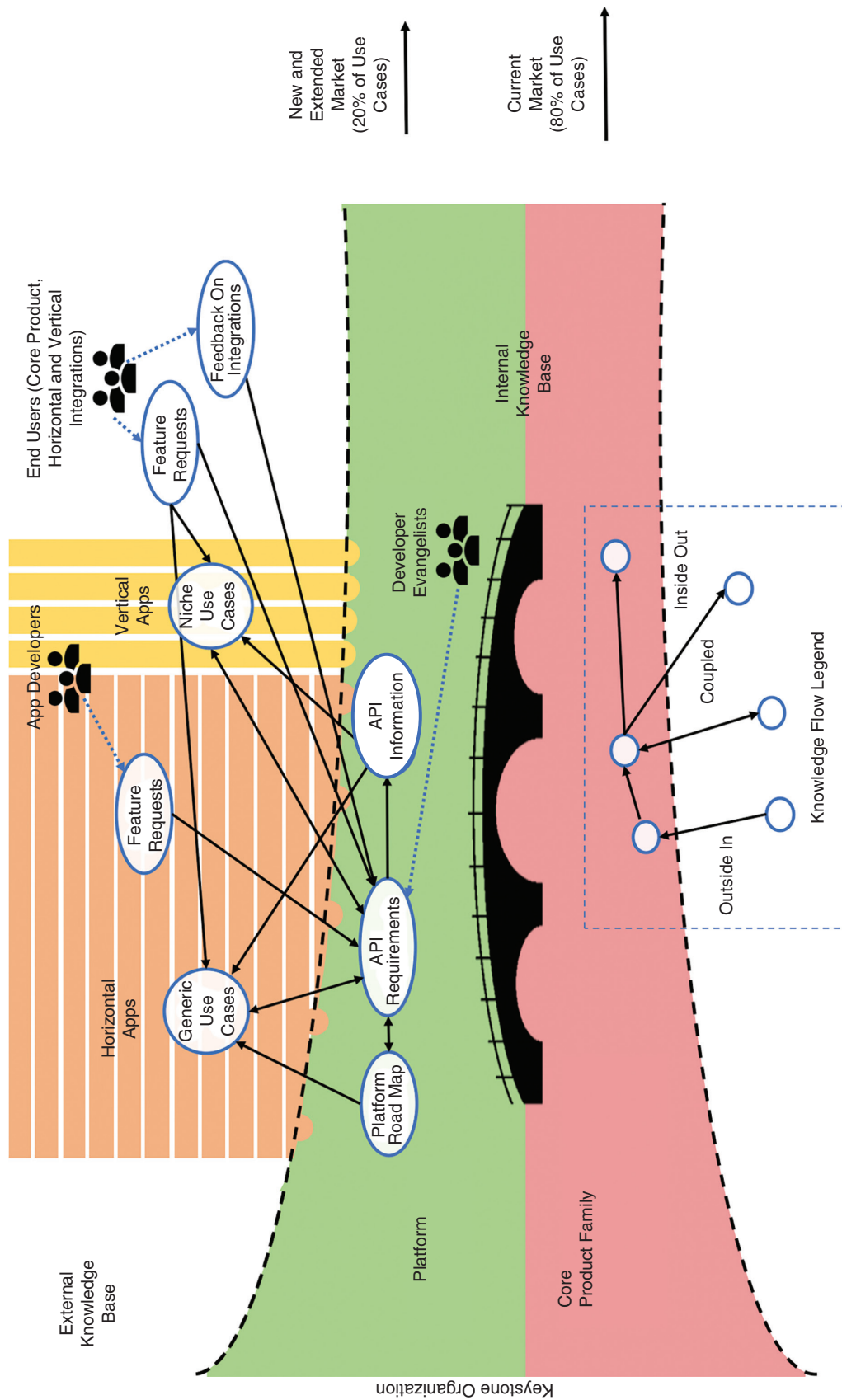


FIGURE 1. Requirements knowledge flows in the SECO, as adapted from the OI model by Chesbrough and Bogers.² The knowledge flow legend indicates the type of knowledge flow represented by the directional arrows.

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—for example, 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 is a risk to the SECO's health as it can damage relationships built over time and the overall value proposition of the platform provider.⁷

2. *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 that 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.
3. *Challenge 3. Growing the core keystone product functionality through acquisitions from the SECO while ensuring customer adoption is demanding:* 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.

4. *Challenge 4. Managing undesired latent requirements while maintaining control within the SECO is difficult:* 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 in terms of 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.

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 previously listed challenges, we distilled this approach from our study participants and how they addressed these overarching issues. 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 techniques 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 context.

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 capabilities¹⁰ 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% represents the use cases that have the widest reach within the market. The keystone opens APIs to enable the complementors to fill the remaining 20%. Figure 1 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:⁸

[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 keystone¹¹ whereby keystones exercise their power wisely¹² and are careful to neither 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.⁵ Two factors to consider in these decisions are 1) a feature's position in the commoditization cycle (for example, innovative to differentiating to commodity product) and 2) the general stance of the platform provider's product compared to competition on the market.⁸

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 theory¹³ 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 and 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 (see the section "Innovation Through Build/Buy/Partner").

<AU: Kindly check that the citation to the section "Innovation Through Build/Buy/Partner" is appropriate.>

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 the sometimes-brutal feedback of the complementors as demanding and technically knowledgeable platform users is a necessary evil (devblog.xero.com/developers-as-your-customer-or-how-i-learned-to-stop-worrying-and-love-the-feedback-loop-19a3caa62aa3) 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.

Innovation Through Build/Buy/Partner

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

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

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 road maps of the keystone and the strategic complementor (similar relationships can also be expected between complementors directly⁴). This tighter alignment can benefit both the keystone and the strategic partners by ensuring that their road maps 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

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

Co-Opetition as Orchestrating Mutual Benefit in Partnerships

The build/buy/partner decisions can contest the 80–20% boundary described in the section “The SECO 80/20 Rule in Selecting Requirements,” and the situation is not without friction at times. <AU: Please check whether changing “it” to “the situation” conveys the intended meaning.>

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, in which the complementor's offering coexists 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 it maintains its strong position in the U.K. market, and by staying in the Xero ecosystem, it has 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 that “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

ABOUT THE AUTHORS



DANIELA DAMIAN is a professor of software engineering in the Department of Computer Science at the University of Victoria, Victoria, British Columbia, V8W 3P6, Canada. Her research interests include empirical software engineering, requirements engineering, and global software development. **<AU: Please provide the highest academic degree/field of study and institution, IEEE Member status if applicable, and a high-resolution author photo.>** Contact her at danielad@uvic.ca.



DAVID JOHNSON is a postdoctoral researcher with the Multimodal Behavior Processing Group at the Center for Cognitive Interaction Technology at Bielefeld University. **<AU: Please check whether the affiliation for David Johnson is the University of Victoria or Bielefeld University in the byline and in the author biography. Please also provide city, state, postal code, and country of current affiliation.>** His research interests include machine learning and human-machine interaction as methods for augmenting human and machine capabilities. **<AU: Please provide the highest academic degree/field of study and institution, IEEE Member status if applicable, and a high-resolution author photo.>** Contact him at davidjo@uvic.ca.



TONY CLEAR is an associate professor in the School of Engineering, Computer and Mathematical Sciences at the Auckland University of Technology, Auckland, Auckland, 1142, New Zealand. **<AU: Please provide research interests, highest academic degree/field of study and institution, IEEE Member status if applicable, and a high-resolution author photo.>** Contact him at tony.clear@aut.ac.nz.



JOHAN LINÅKER is a postdoctoral researcher at Lund University, Lund, Skane, **<AU: Please provide postal code.>** Sweden, and a senior researcher at RISE Research Institutes of Sweden. His research interests include empirical software engineering research in open innovation in the industry and public sectors. **<AU: Please provide the highest academic degree/field of study and institution, IEEE Member status if applicable, and a high-resolution author photo.>** Contact him at johan.linaker@cs.lth.se.



KELLY BLINCOE is a senior lecturer in the Department of Electrical, Computer, and Software Engineering at the University of Auckland. Her research interests include collaborative software development, software analysis, and software requirements. **<AU: Please provide the highest academic degree/field of study and institution, IEEE Member status if applicable, and a high-resolution author photo.>** Contact her at <https://hasel.auckland.ac.nz/> or k.blincoe@auckland.ac.nz.

available to the entire ecosystem. The customer is now presented with a 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; if complementors in the SECO start to experience unfavorable behavior 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, that is, whether to 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, such as forms of data aggregation that introduce privacy or intellectual property risks **<AU: Kindly check that the expansion of IP is correct.>** 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, that is, emerging innovation without negative consequences such as compromised customer privacy.

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

Jansen and Cusumano's classification model¹⁶ situates our ecosystem case organizations 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: SECOs. 

Acknowledgment

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by **<AU: Please provide the review board/committee and/or application number/specific declaration.>**

References

1. P. Simon, *The Age of the Platform: How Amazon, Apple, Facebook, and Google Have Redefined Business*. 2011. **<AU: Please provide the publisher name and location.>**
2. H. Chesbrough and M. Bogers, *Explicating Open Innovation: Clarifying an Emerging Paradigm for Understanding Innovation*. Oxford: Oxford Univ. Press, 2014, pp. 3–28.
3. S. Jansen, "A focus area maturity model for software ecosystem governance," *Inf. Softw. Technol.*, vol. 118, p. 106219, 2020. doi: 10.1016/j.infsof.2019.106219. **<AU: Please provide the issue number or month.>**
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-Inf. Softw. Eng. J.*, 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," *J. Syst. Archit.*, vol. 54, no. 6, pp. 576–593, 2008. doi: 10.1016/j.sysarc.2007.12.004.
6. M. Iansiti and R. Levien, *The Key-stone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*. Harvard Business Press, 2004. <AU: Please provide the publisher location.>
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, A. N. Duc, S. Gao, and G. Sindre, "A systematic mapping study on requirements engineering in software ecosystems," *J. Inform. Technol. Res. (JITR)*, vol. 11, pp. 49–69, 2018. <AU: Please provide the issue number or month.>
9. J. Bosch, *Speed, Data, and Ecosystems: Excelling in a Software-Driven World*. Boca Raton, FL: 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, CA: 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 *Proc. ICEIS*, 2017, vol. 3, pp. 215–226. <AU: Please spell out ICEIS.>
12. G. Valença, C. Alves, and S. Jansen, "A power perspective on software ecosystem partnerships," in *Proc. Int. Conf. Product-Focused Softw. Process Improve*, 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 Innovat. Manag.*, vol. 17, no. 4, pp. 251–258, 2008. doi: 10.1111/j.1467-8691.2008.00499.x.
14. M. Borg et al., "Selecting component sourcing options: A survey of software engineering as broader make-or-buy decisions," *Inf. Softw. Technol.*, vol. 112, pp. 18–34, 2019. doi: 10.1016/j.infsof.2019.03.015. <AU: Please provide the issue number or month.>
15. H. van Leeuwen, "Accounting software minnow says it can prosper in Xero's shadow," *Financial Review*, Feb. 12, 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," in *Proc. IWSECO*, 2012, p. 41. <AU: Please spell out IWSECO.>

Managing the various requirements within the ecosystem brings challenges to the partnerships among the keystone organization and the complementors.

In the keystone organization, innovation lies in the capabilities of the platform to act as a bridge between app developers and the core product family, thus enabling new extensions and integrations in the form of third-party apps.

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.

Managing feedback from third-party app developers as key end users is difficult and requires patience and diplomacy.

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 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.