TOWARDS A CONCEPTUAL FRAMEWORK FOR THE DESIGN, DEVELOPMENT AND IMPLEMENTATION OF TECHNOLOGY PLATFORMS FROM A PLATFORM OWNER'S PERSPECTIVE

HILDE HERMAN
Stellenbosch University, Department of Industrial Engineering, South Africa
17095069@sun.ac.za

SARA S. (SAARTJIE) GROBBELAAR
Stellenbosch University, Department of Industrial Engineering, South Africa AND
DST-NRF Centre of Excellence in Scientometrics and Science, Technology and
Innovation Policy (SciSTIP)
ssgrobbelaar@sun.ac.za

CALIE W.I. PISTORIUS
Stellenbosch University, Department of Industrial Engineering, South Africa AND
DeltaHedron Ltd, United Kingdom
calie.pistorius@deltahedron.co.uk

ABSTRACT

Technology platforms have the potential to provide innovative solutions to real-world problems. Specifically in healthcare, potential solutions include improved access to quality healthcare, availability of data and quicker turnaround times for tests. In order for these platforms to successfully provide such solutions, the platform owner ought to manage its platform and corresponding ecosystem.

This paper reports on the methodology and results of an inventory framework regarding technology platforms. The inventory framework comprises three levels. Each level represents a participant within the technology platform ecosystem: either the platform owner, the app developer or the end-user. The framework provides the key concepts which the platform owner, as keystone firm, might consider for each of the three levels. Within each ecosystem level, similar concepts were categorized together. These categories relate to the software architecture, firm strategy and internal organization, possible entry barriers, control and feedback mechanisms, the importance of usability and considering the context of use. The significance of the platform owner's role in the governance and health of the ecosystem was also discussed.

Keywords: Technology platforms; ecosystem; inventory framework; platform owner; developer; South African healthcare

INTRODUCTION

The increasing connectedness between firms and the potential that collaboration holds motivates many organisations to no longer solely rely on internal innovation, but also on co-innovation within an ecosystem of firms. The analogy drawn between biological and business ecosystems provides useful insights into the nature of these business ecosystems and maintenance of ecosystem health. Business ecosystems research includes work by Moore (1993) and Iansiti and Levien (2004). Criticisms of the ecosystem perspective include arguments that the analogy to natural ecosystems is flawed, the wide-spread use of this perspective lead to unclear definitions of the concept and the absence of ready metrics such as those available for industry clusters (Oh, Phillips, Park & Lee, 2016). Despite these criticisms, the ecosystem approach is still widely recognised as a legitimate perspective (Iansiti & Levien, 2004; Peltoniemi, 2006; Jansen & Cusumano, 2012; Gawer & Cusumano, 2013; Koch & Kerschbaum, 2014).

Specifically in the software industry, firms are connected via a common technological platform and researchers often adopt a software ecosystem (SECO) perspective to understand the operation of these systems (Barbosa & Alves, 2011). Software ecosystems comprise all the actors that function together as a unit to deliver software and software-related services. These ecosystems are frequently enabled by a common technological platform which allows for the exchange of information, resources and artifacts (Jansen & Cusumano, 2012). Software ecosystems can be seen as a subset of business ecosystems (Jansen & Cusumano, 2012).

This paper presents the methodology and results of an inventory framework for platform owners in managing their platform ecosystems. This research draws from several research fields including software ecosystem (SECO) and business ecosystem research from Jansen and Cusumano (2012) and Iansiti and Levien (2004), technology platforms' key concepts from Gawer (2014) and Herman, Grobbelaar and Pistorius (2017), modularity and the architecture of platforms as described by Baldwin and Woodard (2008a) and platform governance and evolution (Tiwana, Konsynski & Bush, 2010). This paper commences with a background literature review on technology platforms and ecosystems, followed by the research methodology. The resulting inventory framework regarding platform ecosystems from a platform owner's perspective is then presented and discussed. The paper concludes with suggestions for future work.

LITERATURE REVIEW

Ecosystem Perspective

The software ecosystem (SECO) definition by Jansen, Finkelstein and Brinkkemper (2009a:35) states that a SECO is a "set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts." The ecosystem participants defined in this research are indicated by the circled components in Figure 1. The remaining components are examples of other possible actors in the SECO (Koch & Kerschbaum, 2014).

In order for the ecosystem to function as a healthy unit, governance of the platform is needed. Although all parties share in the success of the ecosystem, the governance is usually the responsibility of the platform leader or keystone firm (Iansiti & Levien, 2004; Gawer & Cusumano, 2008). This research aims to define the ecosystem into three levels similar to that of Jansen, Finkelstein, *et al*., (2009a). The three levels include the platform owner, the app developers and the end-users. In terms of user and open innovation context, the app developers and end-users are the most important participants of the ecosystem (Koch & Kerschbaum, 2014). These main contributors of open innovation are indicated in the white circles in Figure 1.

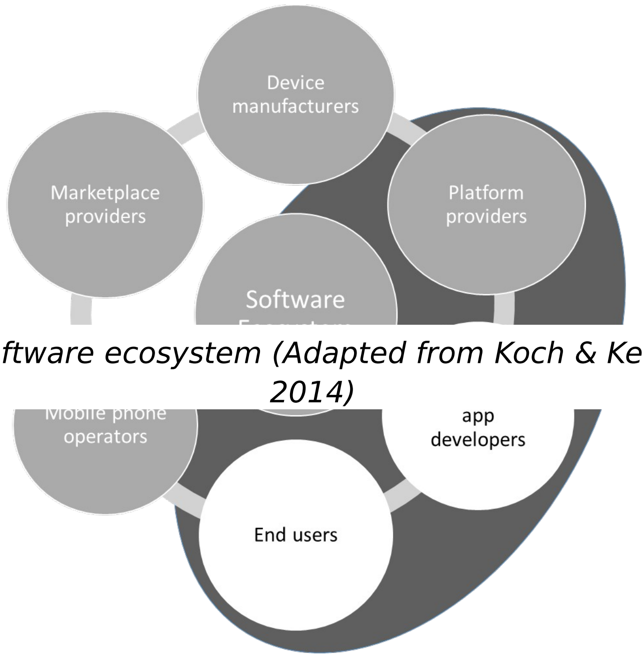


Figure 1: Software ecosystem (Adapted from Koch & Kerschbaum, 2014)

Although sharing several aspects with business ecosystems, software ecosystems differ from it. In software ecosystems, both the actors and their software components can have an effect on ecosystem health. For example, the software component developed by an ecosystem actor can increase the use of the platform (positive effect on health) whilst at the same time the actor itself can negatively influence the ecosystem health due to low productivity and robustness (negative effect on health) (Manikas & Hansen, 2013). The ecosystem referred to in this paper also function around a central software platform resulting in a single entity (platform owner) having the largest governing responsibility within the ecosystem. Similar to business ecosystems, software ecosystems have another major difference which is the ability of conscious choice to cooperate.

Introduction to technology platforms

The term “platform” is used in a range of literature resulting in a variety of definitions describing a platform. Two of the most common perspectives regarding platforms are the engineering or technological view and the economic or transactional view of platforms (Baldwin & Woodard, 2008; Gawer, 2014a). Gawer (2014) differentiates between these views with the first as a modular technological architecture focusing on platform innovation and the second a type of market focusing on platform competition. A platform was conceptualised from an organizational perspective as an evolving organization that coordinates agents who innovate and compete, creates value and entails a modular technological architecture. It should also be mentioned that product platforms (Baldwin & Woodard, 2008) are also seen as a classification of platforms but not deemed applicable to this study.

Despite the different perspectives on platforms, Baldwin & Woodard (2008a) argue that platforms have common roots in engineering design. They state that platform architectures are fundamentally the same and that any platform system comprises of three components namely the evolvable complements, the stable core components and the versatile and flexible interfaces between the two.

Beyond the different platform perspectives, there are a number of classifications of platforms in organisational settings. Gawer and Cusumano (2013) classify organisational platforms into internal and external platforms. Internal platforms operate within a single firm with closed interfaces and are managed by the firm’s managerial hierarchy. An external platform involves industry firms governed by its platform leader and is innovated upon by its complementors who utilise the platform through open interfaces. Moazed (2016) also states that there are many different types of platforms with diverse ways of creating value and these can be divided into two main categories namely maker and exchange platforms. Maker platforms enable third parties to make content on top of the platform while exchange platforms facilitate exchanges between producers and consumers to create value. These categories correspond with the two platform views (engineering and economic views) defined previously. Table 1 aims to clarify these concepts and provides examples.

Table 1: Different types of platforms

Platform perspective	**Platform type**	**Core value being exchanged**	**Examples**
Exchange or transactional platform (linking	Services marketplace	A service	Uber, Airbnb
	Product marketplace	A physical product	Amazon, eBay
	Payment	Payment	PayPal,

with Economic perspective)	platform		Snapscan
	Investment platform	Investment	CircleUp, Prosper
	Social networking platform	Friending interaction model	Facebook, LinkedIn
	Communication platform	Direct communication	Skype, WhatsApp
Maker or technology platform (Linking with engineering perspective)	Closed development platform	Software designed to aquire access to data (via APIs)	Fitbit, Salesforce
	Controlled development platform	Software developed in a controlled, integrated environment	iOS, Windows 8
	Open development platform	Open source software	Linux, Android
	Content platform	Social: discovery and interaction of people. Media: Discovery and interaction of media	Social: Twitter, Instagram Media: YouTube, Amazon kindle

This research defines a platform as a technology that acts as a foundation used by multiple firms and that connects multiple actors together for a common purpose (Gawer & Cusumano, 2002). In the context of this paper, the technology refers to a software platform that enables developers to innovate upon and provide products and services to end-users. The value of such a platform is directly related to the number of complementary products, services and users that adopt the platform (network effects). This supports the importance of maintaining the health and good governance of the software and its ecosystem.

Within such ecosystems, platforms have the potential to address many different challenges in the modern world. One such application is for solutions in the healthcare industry.

Technology platforms in the South African context

The growth of non-communicable or chronic diseases especially in developing countries results in the need for unique healthcare support strategies which may be difficult to deliver (Kahn, Yang & Kahn, 2010;

Assamala, 2014). Africa Health Statistics provide insight into some critical health aspects in Africa (African Union Department of Social Affairs, 2017) including deaths related to infant mortality, neonatal mortality, malaria and the number of citizens receiving treatment for tuberculosis and HIV/Aids. Compounding the urgency of these conditions is the growing need for improved access and quality throughout the public health sector (Harrison, 2009). Technology platforms can provide solutions such as the ability to collect data and communicate information, improved remote monitoring of patients, access to larger databases for better decision-making abilities, monitoring of pharmaceuticals and vaccines, patient self-education which can speed up diagnosis and general improved efficiency and point of care services (Kahn *et al*., 2010; Assamala, 2014).

The resistance of the healthcare industry to the adoption of platforms could be linked to industry specific barriers such as the high cost of failure, sensitive data, resource intensity and the high regulatory control in the healthcare environment (Harvey & Harvey, 2014; Parker, Van Alstyne & Choudary, 2016) In addition to this, healthcare platforms specifically tailored to the African context have not been researched to the same extent as those in developed countries. There is currently no specific framework for implementation in this context known to the researchers.

With the rapid advancement in digital technology and the popularity of mobile devices throughout Africa (GSMA, 2017), the implementation of technology platforms could provide some of the desired solutions (van Heerden, Harris, van Rooyen, Barnabas, Ramanathan, Ngcobo, Mpiyakhe & Comulada, 2017). Therefore the overarching aim of this research is to develop a framework for the design, development and implementation of technology platforms for the Southern African healthcare context which aims to increase the adoption of these platforms as much needed solutions.

Purpose of this paper

The work presented in this paper follows from previous work by the authors including a systematic review conducted to identify the key concepts in technology platform literature (Herman, Grobbelaar & Pistorius, 2017). The results of the systematic review provided the motivation for investigating the platform ecosystem at the platform owner, developer and end-user levels respectively. Therefore, as a continuation from the previous review, this paper presents an inventory framework for each of these three ecosystem levels. This work is the precondition for

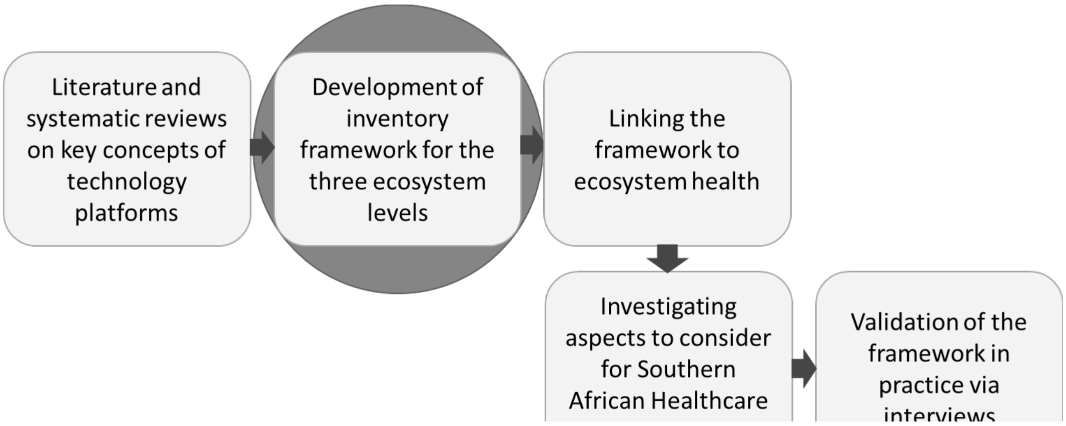

Literature and systematic reviews on key concepts of technology platforms
Development of inventory framework for the three ecosystem levels
Linking the framework to ecosystem health
Investigating aspects to consider for Southern African Healthcare
Validation of the framework in practice via interviews

Figure 2: Context of this paper as indicated with grey circle

future empirical work including validation of the framework in practice. The conceptual framework development process outlined by (Jabareen, 2009) was followed to guide the inventory framework development process. The inventory framework forms a key part in the larger research aim of developing a framework for the design, development and implementation of technology platforms in the African Healthcare context. Figure 2 situates this paper within the larger research project.

METHODOLOGY

In order to develop the proposed framework, the qualitative systematic method outlined by Jabareen (2009) was followed. This process of building a conceptual framework follows a grounded theory approach which aims to "generate, identify and trace a phenomenon's major concepts – each of which has its own attributes, characteristics, assumptions, limitations, distinct perspectives and specific function within the conceptual framework – that shed more light on the phenomenon represented by the concepts themselves" (Jabareen, 2009:53). The advantages of this approach include its flexibility, ease of modification and the focus on understanding instead of predicting. This method develops a framework through a qualitative analysis process of multidisciplinary nature to provide an interpretative approach to social reality (Jabareen, 2009).

A key motivation for adopting the methodology outlined by Jabareen (2009) is the multidisciplinary nature of the research. The different research areas involved with the technology platform governance, management, design and general functioning include software engineering, innovation and management (Van Angeren, Alves & Jansen, 2016), strategy, economics, knowledge management and organisational studies literature (Gawer, 2009). In their systematic review aiming to identify key concepts in the technology platform literature, Herman, Grobbelaar and Pistorius (2017) also discussed the diversity of literature regarding technology platforms which includes studies on innovation, social science, systems perspective and nature related fields. The research methodology is clarified in Table 2.

Table 2: Research Methodology

Phase in this paper	**Link to Jabareen, (2009)**	**Objective of phase**	**Research approach**
Phase 1: Mapping data sources	Phase 1: Mapping data sources	Map spectrum of multidisciplinary literature regarding question.	Establish three levels and acquire level-specific data
Phase 2: Reading,	Phase 2: Reading and	Read selected data and categorise by discipline and	Read through

concept identification and categorization of data	categorizing of data	scale of importance.	level-specific data sources and extract relevant concepts from each level.
	Phase 3: Identifying and naming concepts	Read and re-read data to discover concepts. Allow for concepts to emerge from literature.	
	Phase 4: Deconstructing and categorising concepts	Identify the main attributes, characteristics, assumptions and role or each concept. This is followed by categorising the concepts accordingly.	
Phase 3: Framework development	Phase 5: Integrating concepts	Integrate and group together similar concepts to form one concept. This phase reduces the number of concepts.	Conceptualise a framework.
	Phase 6: Synthesis, resynthesis	Synthesise concepts into a framework. This is an iterative process and includes repetitive synthesis and resynthesis.	
Phase 4: Future work	Phase 7: Validation	To establish whether the framework makes sense to others, not only researcher. Includes presenting, receive feedback.	The framework will be validated as part of the future work as indicated in Figure 2.
	Phase 8: Rethinking framework	A multidisciplinary framework will always be dynamic and needs to be revised.	

Phase 1: Mapping of data sources

In a software ecosystem context, Jansen, Brinkkemper and Finkelstein (2009) and Van Angeren, Alves and Jansen (2016) include three ecosystem participants: the platform owner, app developers and customers. Herman, Grobbelaar and Pistorius (2017) also state that technology platform literature often adopts the viewpoint of one or more of these three ecosystem participant “levels” when discussing platforms. As a result, this paper includes platform owner, the app developers and end-users as the software ecosystem. Iansiti and Levien (2004) argues that the overall health and evolution of such an ecosystem is dependent on managing the complex relationships between the participants. The ecosystem builds around the technology platform and consequently results in the platform owner taking up a large portion of the responsibility for maintaining ecosystem health (Jansen, Brinkkemper, *et al*., 2009). The multi-faceted relationships in these ecosystems need to be managed for ecosystem health (Van Angeren, Alves & Jansen, 2016) as owners not only depend on their own platform for survival, but also on the extensions and applications within their ecosystem.

Phase 1 in the research methodology required the mapping and identification of all relevant multidisciplinary literature in order to develop the framework. The approach was to investigate the requirements of each of the three levels in order to identify the issues faced and the context in which each of the level specific participants operate. This would provide a comprehensive picture of the ecosystem required to develop the framework. The search was conducted using the Google Scholar, Scopus and Research Gate databases. The search terms used were level-specific in order to identify studies that would render sufficient information regarding each level. Other than the specific descriptions of each level, other factors guided the level-specific search. The background literature review provided some insight into the roles and context of each level and how they relate back to the platform owner. Consequently, certain topics regarding each level could be formulated and used as a starting point for the level specific search. Table 3 indicates the descriptions, the topics derived from the background literature review and how these were used as guidelines in the search process.

Table 3: Ecosystem levels, descriptions and example search starting points

Ecosystem level	**Description of level**	**Topics from background literature review**	**Examples of how topic was translated to guide search**
Platform owner	The platform owner owns and manages the software platform and its boundary resources. It is also usually responsible for the governance of the ecosystem forming around the platform.	Responsible for the software architecture	What does a platform architecture look like?
		Responsible for ecosystem governance	What does ecosystem governance entail?
Developer	The app developer is the actors either within the platform owner company (internal platform) or third-party	Platform owner regulates developer's ability to develop	How can the platform owner enhance the developers' ability to

	companies (external platform) who build components on top of the platform.	innovative products/services.	innovate?
		The platform owner determines entry barriers to the ecosystem.	What entry barriers and motivations for joining the ecosystem can the platform owner control?
End-user	The app users are the end-users of the applications developed using the platform.	Network effects	How can the platform owner aid in motivating end-users to adopt products/services developed on their platform?
		Importance of ensuring usability	

Phase 2: Reading, concept identification and categorisation of data

Phase 2 comprised an extensive reading and re-reading (Jabareen, 2009) of the identified literature for each of the three levels namely the platform owner, developer and the end-user. The aim was to allow for general understanding of the levels and for concepts to emerge from the literature for use in the development of the framework. Each of the three levels was investigated and their environment and characteristics established in order to provide insight. The main concepts for each level were determined. Figure 3 clarifies the three levels considered and their relation to one another. In commercial apps, the marketplace fulfils a key role in the success of the app. Although the marketplace dynamics are considered in this paper, it is not a part of the scope of this this research.

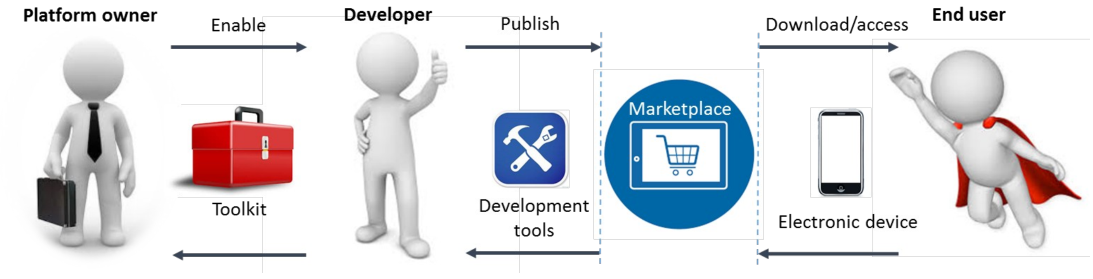


Figure 3: Three levels' context

Phase 3: Framework development

For the formulation of the inventory framework, the focus was to integrate and group together the concepts identified in Phase 2 to reduce the total number of concepts (Jabareen, 2009). These concepts were then synthesised into a framework consisting of an inventory of concepts to consider at each level from the perspective of a platform owner. The inventory framework was constructed as a result of the context given for each level through the read and re-reading of the various literature sources in Phase 2.

RESULTS

Following the level-specific literature search, concepts were identified, grouped and synthesised into the inventory framework. The aim of the framework is to inform the platform owner of concepts regarding the three levels and consequently result in better platform management. Hence, the concepts included can be translated into actionable activities for the platform owner itself. The concepts draws from diverse areas such as business strategy (Gawer & Cusumano, 2002), software engineering (Wasserman & Fosser, 2010; Joorabchi, Mesbah & Kruchten, 2013) innovation management (Scholten & Scholten, 2012; Koch & Kerschbaum, 2014) social science (Lima, Santos, Oliveira & Werner, 2016), software ecosystems (Jansen, 2014) and business ecosystems (Iansiti & Levien, 2004; Den Hartigh, Tol & Visscher, 2006; Rong, Lin, Shi & Yu, 2013) It should be mentioned that some concepts are broad and could potentially be researched further. Also, some of the included concepts may overlap. Due to the variety of platforms, the inventory was developed to be as general as possible. The exact scope of each platform will vary (internal vs external platform for example) and all included concepts may not always be applicable for every platform owner.

The framework can be approached by asking the following questions for each level:

I. Platform owner level: What should the platform owner consider with regards to its own firm and the software platform?

II. Developer level: What should the platform owner consider with regards to the developers using their platform?

III. End-user level: What should the platform owner be aware of and apply to its platform to enhance the development and uptake of end-user products and services?

Appendix A gives more information on all concepts included in the inventory framework and provides the references.

Level 1: Platform owner

As the name suggests, the platform owner is the provider and manager of the central software platform. The platform owner can be seen as the platform leader (Gawer & Cusumano, 2013) or keystone firm (Van Angeren *et al.*, 2016). The platform owner is responsible for the underlying technology upon which the ecosystem operates as well as the openness thereof (Boudreau, 2010) (Jansen, Brinkkemper, Souer & Luinenburg, 2012).

In an attempt to view the many seemingly different uses of platforms in a unified view, Baldwin & Woodard (2008a) argues that there is a fundamental similarity at the architectural level of platforms. The architecture of a platform can be defined by its ability to allow certain architectural components to remain fixed, while other components change over time. It thus implies that the platform architecture firstly comprises of stable and reusable core components with low variety but high reusability and secondly, it comprises of peripheral components that are variable and have a high variety but low reusability (Boudreau, 2010) (Baldwin & Woodard, 2008). These are generally referred to as the complements of the platform (Baldwin & Woodard, 2008) (Gawer & Cusumano, 2008). The platform and its complements interoperate through the predetermined interfaces. By splitting the system into these modular components, the design and production across the platform are split across multiple firms allowing for more innovation and specialization to occur (Gawer, 2014b).

This fundamental modular architecture of platforms in the context of this research is explained in Figure 4. The platform owner provides the stable core platform with high reusability, but low variety. On top of this modular platform, the developers (complements) build their own variety of applications (apps) resulting in new innovations. This can be linked to the health of the ecosystem seeing diversity as one of the requirements of a healthy ecosystem (Den Hartigh, Tol & Visscher, 2006). The interfaces involves the boundary resources (Software development kit (SDK), application programming interfaces (API), etcetera) and determine the ease of use and integration of the core platform with developers. These interfaces reflect the decisions regarding platform openness which the platform owner needs to establish (Parker & Van Alstyne, 2013). Through adopting this architectural design, the core platform can be re-used without extensive modification to produce a variety of applications. The platform owner is also responsible for establishing standards, licencing

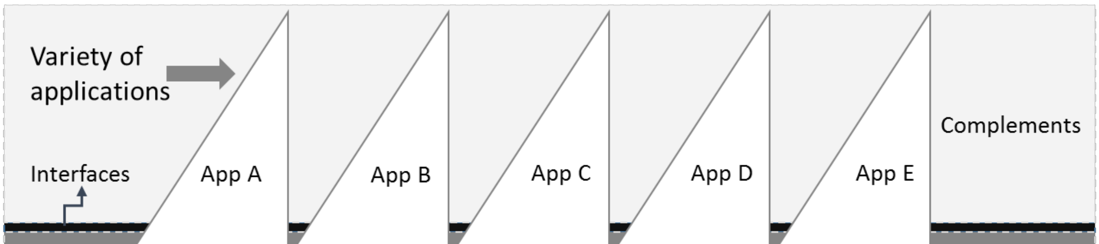


Figure 4: Platform architecture conceptualisation

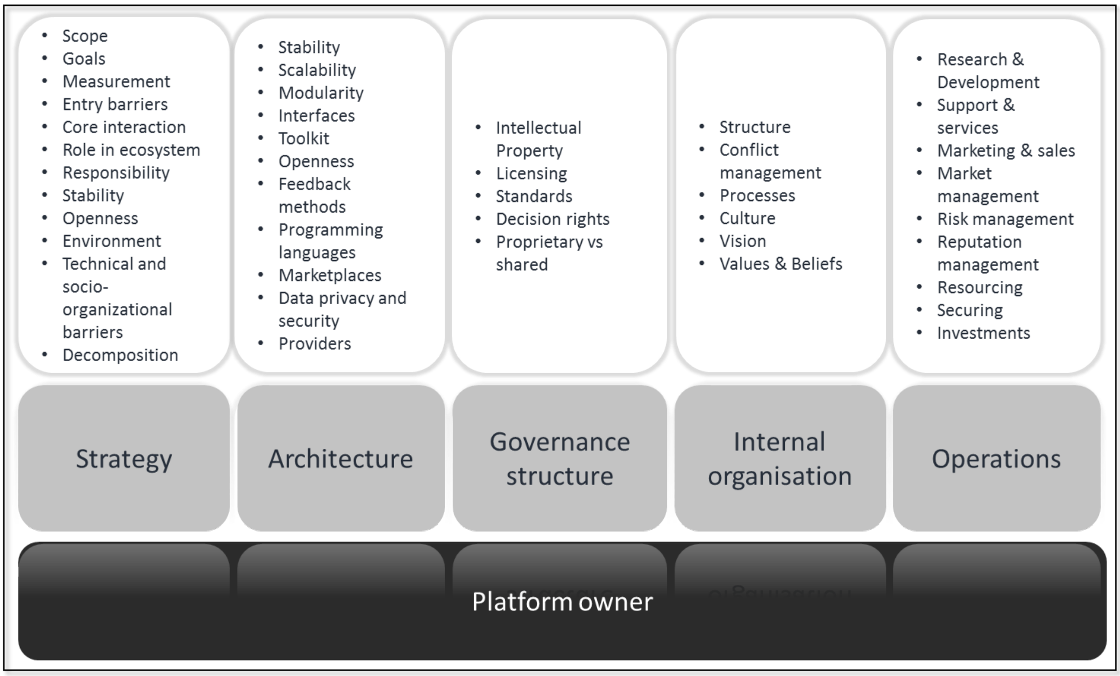
requirements and the distribution of decision rights within the ecosystem (Tiwana, Konsynski & Bush, 2010).

Apart from the technical aspects regarding the software platform and its architecture, the platform owner should make vital decisions regarding the functioning of its firm and its customer base. This includes explicitly stating the scope of its firm as well as determining the envisioned ecosystem participants and outlining its internal and external environments (Berk, Jansen & Luinenburg, 2010). Baars and Jansen (2012) distinguish between governance and governance structure where governance refers to the processes, procedures and tools that the leader uses to execute their strategy and the governance structure refers to the corresponding responsibility, control and measurement. The platform owner should be aware of its governance and governance structure as the health of the ecosystem is directly linked to successful governance (Jansen, 2014).

The first level of the resulting inventory framework shown in Figure 5 focuses on aspects that the platform owner should consider within its own firm and as the designer of the software platform. From the level-specific literature, it can be concluded that the platform owner has three main responsibilities: (1) to maintain the health and reputation of its own firm as the keystone of the ecosystem (Berk, Jansen, Luinenburg & van den Berk, 2010; Jansen *et al*., 2012; Koch & Kerschbaum, 2014) (2) to provide the software platform and establish its openness (Baldwin & Woodard, 2008; Boudreau, 2010; Rong *et al*., 2013) and (3) to govern the ecosystem (Tiwana *et al*., 2010; Jansen & Cusumano, 2012; Gawer, 2014b). The main categories within the inventory framework include strategy, the governance structure, internal organisation of the platform firm, architecture-related concepts and operational measures.

Figure 5: Inventory framework Level 1: Platform owner

The strategy category refers to the guiding principles and important decision-making factors related to the platform owner and its software



platform. The platform owner should explicitly define its scope in terms of its desired ecosystem participants, the core interaction of the platform, the platform openness, the complete software delivery model and also what is excluded from the ecosystem scope (Berk, Jansen, Luinenburg, *et al*., 2010; Parker *et al*., 2016). The concepts related to the software of the platform were categorised within the architecture category. As described previously, the stability, scalability and modularity of a software platform are significant factors as complementary products and services will be developed on top of the platform (Baldwin & Woodard, 2008; Rong *et al*., 2013). These development processes are enabled by the platform interfaces such as software development kits (Gawer & Cusumano, 2013) . The control over these interfaces is linked to the control of the platform and platform evolution (Baldwin & Woodard, 2008).

The second category is the governance structure of the platform firm which refers to the distribution of rights and the rules and protocols within the ecosystem. This includes the development of licencing agreements, the decisions regarding ownership of the platform and the decision rights amongst the ecosystem participants (Tiwana *et al*., 2010; Scholten & Scholten, 2012). Both external and internal ecosystem parties can look to the platform owner (keystone) as a reflection of the ecosystem health and dynamics (Jansen, Brinkkemper, *et al*., 2009; Berk, Jansen, Luinenburg, *et al*., 2010). Therefore, the internal organisation section includes the generation and implementation of corporate values, beliefs and culture to promote a healthy firm and ecosystem (Berk, Jansen, Luinenburg, *et al*., 2010; Koch & Kerschbaum, 2014).

The operations category refers to the aspects related to the smooth functioning of the platform firm and its ecosystem. Especially in the information technology (IT) industry, the rate of new emerging technologies requires constant research and development (R&D) efforts (Jansen *et al*., 2012). The platform owner should involve the complete ecosystem and should monitor competing ecosystems. An effective marketing and sales approach could encourage the growth of the ecosystem and Gawer & Cusumano (2002) also places market management as one of three foundations of ecosystem competition. In order for the platform to continuously improve and evolve, the platform owner should seek new opportunities with regards to its scope and ecosystem participants. Ghazawneh & Henfridsson, (2013) refer to this as resourcing. Some risks may be required to evolve the ecosystem. The platform owner can share the intentional risks (monetary investments or adjusting scope for example) within the ecosystem (Gawer & Cusumano, 2014). To minimise the unintentional risks, the owner should continuously adjust the control strategy across the ecosystem (Scholten & Scholten, 2012).

Level 2: App developer

The app developers are the main sources of open innovations within the ecosystem and can be referred to as user innovators (Koch & Kerschbaum, 2014). Depending whether the platform firm is an internal or external platform firm (Gawer & Cusumano, 2013), the app developers either function within the platform firm itself, or as external firms using the platform to develop their own products and services. In the fight to attract more users to the ecosystem, referred to as the "ecosystem war", Ryu, Kim & Kim (2014) emphasise the importance of platform providers to develop positive relationships with third-party developers. In the development of their boundary resources model, Ghazawneh & Henfridsson (2013), state that platform owners should not only focus on developing apps, but invest more into providing the resources for third-party developer support. This support will differ with regard to the type of app the developer aims to produce. There are three broad categories of apps namely native, web-based and hybrid. Native refers to apps operating on a specific device's OS, web-based apps run on a web-browser and hybrid apps are web apps functioning within a native browser (Joorabchi *et al*., 2013).

In order to provide a better platform environment for the developers, platform owners need to be aware of the challenges that developers face and invest into understanding the environment in which they operate. Through this understanding, platform owners can work towards reducing these challenges and thereby possibly increase their customer base and developer loyalty to the platform. Each platform has different standards, expectations, programming languages, Human Computer Interactions (HCIs), toolkits, other support tools and requirements related to human interface aspects (Joorabchi *et al*., 2013). This result in constant variation for developers using different platforms. Adding to these challenges, developers also have to deal with fragmentation related to device memory, CPU speed and graphical resolutions of various devices (Joorabchi *et al*., 2013). Another challenge app developers face is the monitoring and testing of their apps (Joorabchi *et al*., 2013). The platform owner should could provide customer support and feedback mechanisms for developers using their platform.

Apart from being aware of the challenges that the developers face, another important aspect to consider is the motivations behind why developers choose to join a particular ecosystem. Software development kits (SDKs) have been determined as one of the major aspects attracting external developers as they are critical for developing quality applications and can directly be related to platform satisfaction and credibility (Koch & Kerschbaum, 2014; Ryu *et al*., 2014). Ghazawneh & Henfridsson (2013) developed a boundary resources model for platform owners in an attempt to guide the functioning of the software and regulations at the platform interfaces. These interfaces are key in the relationship between the platform owner and app developers. The platform owner should invest in boundary resource design and should modify or constantly develop new boundary resources as a response to feedback and environmental or market changes.

A major reason behind developer adoption of a platform is the openness strategy of the platform. Anvaari & Jansen (2010) define this as “the degree to which a platform supplier allows the platform users to interact with the platform, view, extend or change its components and depends on different technical and commercial aspects such as platform architecture, platform accessibility, platform transparency, licensing state, marketing policy, etcetera”. Therefore the openness strategy of a firm not only includes the architectural aspects, but also the softer aspects reflecting the reputation and values of the platform owner.

The entry barriers to join a software ecosystem is important to establish as a platform owner (Berk, Jansen, Luinenburg, *et al*., 2010; Koch & Kerschbaum, 2014; Van Angeren *et al*., 2016). The entry barriers can be intentionally formulated to ensure quality within the ecosystem, or unintentionally such as its reputation, value creation and market size.

The second level of the inventory framework, shown in Figure 6, presents the concepts that the platform owner should consider with regard to the developers using their platform. The main categories for this level include the entry barriers, the governance structure, the ecosystem related concepts, the concepts related to the platform’s architecture, standards and current models that could be helpful, the different customer support concepts and the control that the platform owner should consider.

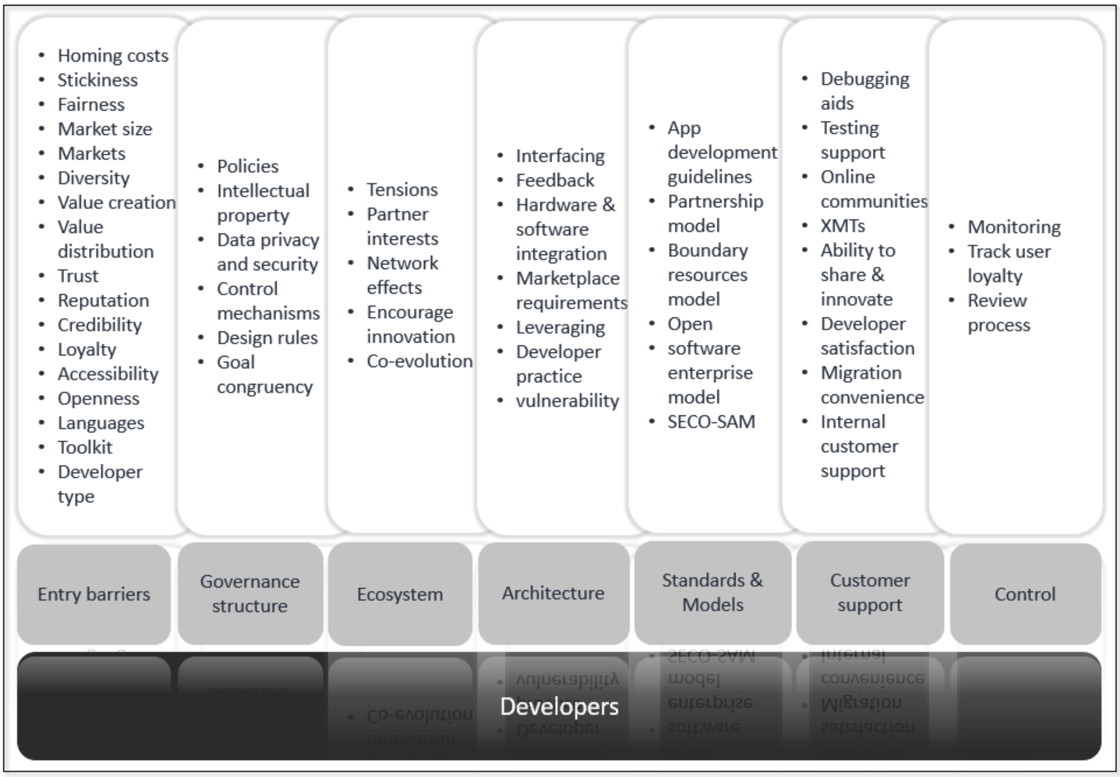


Figure 6: Inventory framework: level 2: Developers

The first category includes the entry barriers of the platform and its ecosystem (Jansen & Cusumano, 2012; Koch & Kerschbaum, 2014). These can be related to both the platform firm itself as well as the software platform and its architecture. The governance structure category refers to the distribution of rights and the rules and protocols within the ecosystem, specifically with regard to the developers, their products and services and the platform-related rules (Cusumano & Gawer, 2002; Tiwana *et al*., 2010). The ecosystem related concepts refer to the elements that involve more than one ecosystem participant and their relationships (Den Hartigh *et al*., 2006; Scholten & Scholten, 2012; Parker & Van Alstyne, 2014a). The architecture category includes specific concepts that the platform firm should be aware of regarding the use of their platform and its interfaces (Mansfield-Devine, 2012; Ryu *et al*., 2014) (Koch & Kerschbaum, 2014). During the reading of the level-specific literature, specific standards and models featured that could provide guidelines for the platform owners in terms of openness decisions and partnerships with other ecosystem participants (Berk, Jansen & Luinenburg, 2010; Ghazawneh & Henfridsson, 2013).

As the developers might not be aware of the internal functioning of the platform, the platform owner should provide sufficient support and testing components to enable developers to successfully use their platform (Joorabchi *et al*., 2013). It might also be required that the platform owner personally engage with the developers. The last category within the developer level of the inventory framework relates to control aspects the platform owner should administer. This includes monitoring of developer performance (Lima *et al*., 2016), tracking their loyalty towards the platform as an indication of more attractive competitor ecosystems (Jansen, 2014) and possibly reviewing the products and services developed on their platform (Nayebi, Desharnais & Abran, 2012).

Level 3: End-user

The end-users in the context of this paper refer to the parties using the products and services developed via the platform. The extent to which the platform owner has an influence on the end-user varies on the type of platform. As mentioned previously, platforms can either be internal or external, depending on their level of openness (Gawer & Cusumano, 2013). Hence, if the platform firm adopts an internal approach, it will have a high level of interaction with and influence on the end users. If the firm adopts an external platform strategy the level of interaction with the end-users will be considerably less. The end-users do however form a part of the ecosystem and therefore the platform owner, regardless of its approach, should be aware of their roles.

In end-user-specific studies, the two most frequently discussed concepts were the usability of the apps (Harrison, Flood & Duce, 2013; Zapata, Fernández-Alemán, Idri & Toval, 2015) and user and usage feedback (Harman, Jia & Zhang, 2012; Bhih, Johnson & Randles, 2016). Usability is becoming a critical factor due to the increase of software technologies being used to perform everyday tasks as well as increasing competition within the software industry. High usability increases user productivity, performance, safety and quality experienced during use (Seffah, Donyaee, Kline & Padda, 2006).

One of the ways in which usability has been defined is as“ [t]he extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (ISO 9241−11, 1998) (Seffah *et al*., 2006). There are different models and standards providing different metrics and attributes of usability. Some usability attributes include effectiveness, efficiency, satisfaction, learnability, memorability, error rate of the system (Nielsen, 1994) and cognitive load (PACMAD in Harrison *et al*., (2013)). Leading operating system (OS) firms also provide their own usability and app design guidelines and state how to incorporate these during development and testing of applications (Android, n.d.; Apple Inc., n.d.).

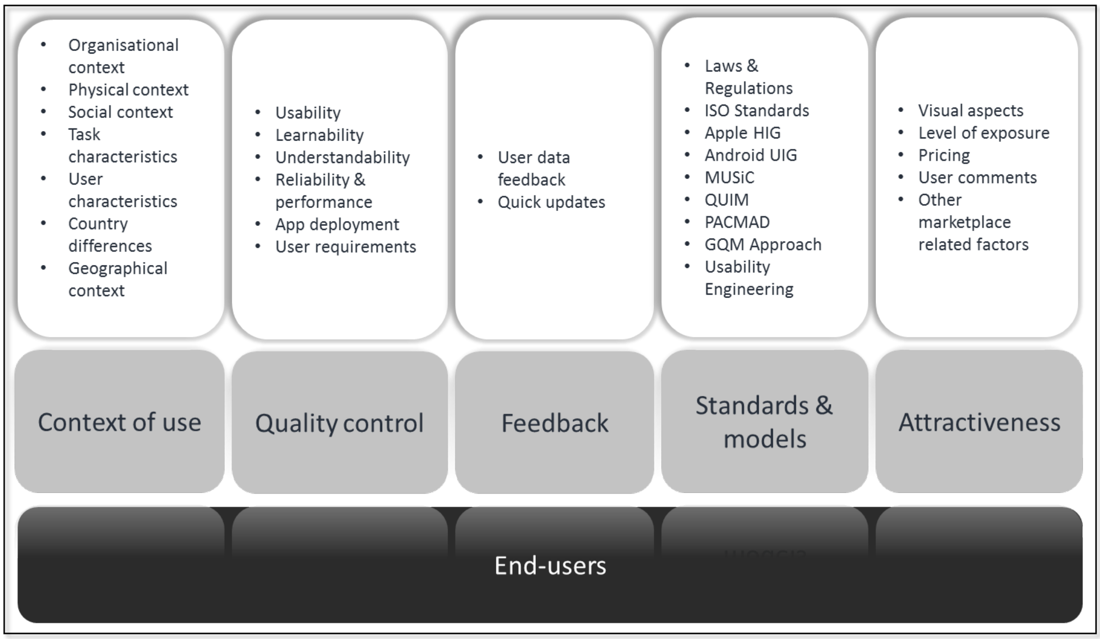
The reasons for purchasing as well as abandoning apps could also provide end-user insight (Lim, Bentley, Kanakam, Ishikawa & Honiden, 2015). Reasons for app abandonment include the availability of better alternatives, being bored with the app, the app crashing or being too slow, the app not being user friendly and the app not having the features as required by the user (Lim *et al*., 2015). The reasons found within the level-specific literature sources regarding the buying or adoption of specific apps included attractiveness factors and how well the app met the user needs (Pagano & Maalej, 2013). Attractiveness aspects included pricing, app description, the app name and icon, the size of the current user base and user reviews of the app (Lim *et al*., 2015).

The user behaviour data can be obtained through different methods. Methods of data acquisition include mining of marketplace data, incorporating user feedback mechanisms within the app software, surveys and app testing methods (for example in a controlled environment) (Pagano & Maalej, 2013; Lim *et al*., 2015) Even if the platform owner does not develop apps within its own firm (in other words an external platform), the platform owner should aim to understand the end-user requirements to be able to provide a better software platform and guidelines for the developers using their platform.

The third level of the inventory includes the concepts related to the end-users. Figure 7 presents this level of the framework. Interpretation of this level depend on whether the platform firm functions as an internal or external platform firm. The overarching categories includes the context of use of the product or service, measures of quality control, feedback mechanisms, standards and models that can aid in app development and other attractiveness related aspects.

As shown in Figure 7, the developer should be aware of the context of use of the app. This not only refers to the physical or social contexts, but also to the geographical context (related to network strength availability) and countries of use (laws and regulations may differ) (Seffah *et al*., 2006). Certain measures of quality control could be established, as the platform owner is indirectly affected by the success of the end-product (Jansen, Brinkkemper & Finkelstein, 2009). As mentioned previously, the motivations regarding end-users’ adoption of products and services are important as they could give firms a competitive edge (Lim *et al*., 2015).

Figure 7: Inventory framework level 3: End-user



The three inventory levels provide specific aspects that a platform owner should consider in the design, development and implementation of its platform. The motivations for a platform owner to invest into translating these concepts to actionable activities can be summarized as follows: (1) external firms might look to the platform owner as a reflection of the ecosystem and the firm’s perceived success and health are therefore motivating aspects for others to join the ecosystem; (2) strategy and architecture related decisions will translate into ecosystem entry barriers; (3) the platform owner carries a large responsibility within the ecosystem and if explicit boundaries, strategy, vision and scope aspects are not established, the evolution of the ecosystem could be in an uncertain direction; (4) the health of firms within the ecosystem is not only dependent on their own actions, but is affected by the complete ecosystem of firms.

CONCLUSION AND FUTURE WORK

This paper presented the process and results of developing an inventory framework for platform management within its platform ecosystem. The framework development process outlined by Jabareen (2009) was followed. As a result, the framework was developed with three specific ecosystem levels namely the platform owner, the developers and the end-users. The concepts within each level were categorised and discussed and can be translated into actionable activities and questions to ask as a platform owner. This framework provides the foundation for the umbrella larger research project regarding the design, development and implementation of technology platforms in the Southern African healthcare context.

Proper management of SECOs can lead to better use of resources, lowering risks, helping a company reach its goals and ultimately result in an increase in revenue (Baars & Jansen, 2012). Baars & Jansen (2012) also encourage research similar to their ecosystem governance framework. As the three level inventory framework is developed from a software ecosystem perspective, future work could include relating this framework to the managing the ecosystem and maintaining its health. Ecosystem health can be defined simplistically as longevity and propensity for growth (Jansen, 2014). Within their ecosystems, platforms also have the ability to shift value from the firm level to the network level as value is co-created by all actors in the ecosystem which substantiates the need to look beyond the firm level to the ecosystem level. The ecosystem as defined in this paper was analysed predominantly at the firm level where each actor was investigated.

Scholten & Scholten (2012:175) found it crucial for the platform owner to perceive the platform and its ecosystem as well as its evolution as a requirement for internal and external innovation. They define innovation management in platform businesses as “an ongoing process that (a) systematically identifies, evaluates and defines the strategic innovation goals of platform and ecosystem evolution; (b) implements innovation strategy both within the company and within the platform ecosystem, and finally monitors and controls strategy implementation.” This definition substantiates the idea to link the framework to ecosystem evaluation as well as exploring the nature and drivers of its evolution.

As this research forms part of a larger research project regarding technology platform implementation in Southern African healthcare, future work would also include determining the healthcare and country-specific related aspects to consider when developing a technology platform specifically for this context. The final framework for design, development and implementation of these platforms in South African healthcare will also be validated in practice through the conducting of interviews. This will form Phase 4 of the methodology outlined in this paper. The inventory framework presented in this paper is the first iteration of concepts to consider as a platform owner and the investigation into healthcare in South Africa as well as industry validation will call for further iterations.

APPENDIX A

Platform owner			
Category	Concept	Description	Key references
Strategy	Goals	The firm should establish the long-term and short-term goals for the firm and the ecosystem and align their partners accordingly.	(Jansen *et al*., 2012)
	Openness	This refers to the amount of information shared about the platform to outsiders. Openness can refer to the extent to which the platform owner decides to make platform aspects available such as open standards, open formats and open source.	(Berk, Jansen, Luinenburg, *et al*., 2010) (Koch & Kerschbaum, 2014)
	Measuremen t	A measurement initiative could be established such as defining KPIs and continuously monitoring the current state of the firm and its ecosystem as well as predicting its future states.	(Baars & Jansen, 2012)
	Responsibilit y	The responsibility of the participants within the ecosystem should be predetermined/established as it will form part of the decision criteria as to why actors choose to join the ecosystem.	(Tiwana *et al*., 2010)
	Entry barriers	Entry barriers enable a sense of control regarding who enters the ecosystem. These barriers can both be too high (risking possible innovation) and too low (reducing quality).	(Berk, Jansen, Luinenburg, *et al*., 2010) (Koch & Kerschbaum, 2014) (Van

			Angeren *et al*., 2016)
	Core interaction	By establishing a core interaction, it establishes focus and it also allows for specialization of the firm in terms of this core interaction.	(Van Alstyne, Parker & Choudary, 2016)
	Scope	Identifying the scope refers to identifying the target market, the product itself, the software delivery model and expectations. An Industry capability stack (Cusumano & Gawer, 2002) is a useful tool to define what the firm will and will not do. The platform owner should be explicit when defining both the ecosystem and its governance strategy.	(Berk, Jansen & Luinenburg, 2010) (Gawer & Cusumano, 2002) (Baars & Jansen, 2012)
	Role in ecosystem	The platform owner should establish its role within the ecosystem and should be vigilant against leaning towards a dominator strategy.	(Iansiti & Levien, 2004)
	Stability	The organisation should have a sense of stability regarding direction, company vision and strategy as well as its relationships with employees and ecosystem actors. The reason for this is that it influences how external actors perceive the platform firm and therefore the ecosystem.	(Berk, Jansen & Luinenburg, 2010)
	Define external environment	The ecosystem does not function in a bubble and the platform owner should be aware of its complete environment including the suppliers, its competing ecosystems, stakeholders and laws and regulations that are applicable to the ecosystem.	(Berk, Jansen, Luinenburg, *et al*., 2010)

	Project distribution barriers	Within the defining of the scope and target participants, the platform owner should consider the technical and socio-organizational barriers for coordination and communication in projects that are geographically distributed.	(Barbosa & Alves, 2011)
	Decomposition	Decomposition refers to the hierarchical decomposition of the ecosystem into subsystems.	(Tiwana *et al*., 2010)
Governance structure	Licensing	The licensing agreements that the developers have to agree to in order to gain access to the platform.	(Scholten & Scholten, 2012)
	Intellectual property (IP)	IP rights should be established in such a manner as to facilitate third party innovation and not limit it.	(Gawer & Cusumano, 2002) (Parker *et al*., 2016)
	Standards	Standards facilitate interoperability between organisations, devices, technologies or data formats. The necessary standards should be enforced by the platform owner.	(Iansiti & Levien, 2004)
	Decision rights	It is important to establish which actors in the ecosystem has authority in decision making and to what extent.	(Tiwana *et al*., 2010)
	Proprietary vs shared	This refers to the ownership of the platform (not to the architectural openness of the platform software).	(Tiwana *et al*., 2010)
Internal organisation	Structure	The actual structure of the platform firm should be carefully planned. For example separating the groups dealing with customer support and those developing complementary products on the platform.	(Gawer & Cusumano, 2002)

	Conflict management	The tensions and conflict that could arise within the platform firm. These situations can occur when assisting external developers with their development process while internal groups are also developing complements.	(Gawer & Cusumano, 2002)
	Processes	These internal processes refer to set processes within the firm such as having meetings with different firm levels to set goals and develop strategies. These might also include processes for dealing with conflicts that might arise in different units of the firm.	(Gawer & Cusumano, 2002)
	Culture	A culture of debate and discussion should be encouraged as it results in dealing with conflicts in a healthy manner and also encourages innovation.	(Koch & Kerschbaum, 2014) (Gawer & Cusumano, 2002)
	Vision	The vision refers to the envisioned future state of the firm and/or ecosystem which should be articulated. It could encourage actors to enact in this vision. Not only for the platform owner, but could also to motivate external actors sharing the vision to join the ecosystem.	(Berk, Jansen & Luinenburg, 2010) (Gawer & Cusumano, 2014)
	Values & Beliefs	The platform owner should establish its values and beliefs and aim to foster common values and beliefs within the other ecosystem participants.	(Tiwana *et al*., 2010)
Architectu re	Scalability	The platform should be designed to adapt to user demand fluctuations. The firm could also leverage the ecosystem capabilities to add to scalability and adaptability.	(Gawer & Cusumano, 2002)

	Stability	Stability specifically refers to the stability of the platform architecture and interfaces. As the connecting point with the developers, the interfaces should be stable.	(Gawer & Cusumano, 2013) (Barbosa & Alves, 2011) (Baldwin & Woodard, 2008)
	Modularity	Modularity entails how certain components of a system can change without affecting the rest of the system. A modular architecture is recommended as it facilitates innovation.	(Tiwana *et al*., 2010) (Cusumano & Gawer, 2002) (Gawer & Cusumano, 2014) (Baldwin & Woodard, 2008)
	Interfaces	The use of standard interfaces such as APIs to standardise how different modules access and interact with the platform.	(Tiwana *et al*., 2010) (Iansiti & Levien, 2004) (Baldwin & Woodard, 2008) (Gawer & Cusumano, 2013)
	Toolkit	The toolkit (such as SDK) is the interface software provided by the platform owner that enable the developers to successfully innovate upon their platform.	(Koch & Kerschbaum, 2014) (Ryu *et al*., 2014) (Ghazawneh & Henfridsson, 2013)
	Openness	The level openness of the architecture should be decided upon by the platform owner. It refers to the extent to which outsiders can access the different layers of the platform and change some of the functionality for their own use. Openness is a balance between allowing developers to effectively develop quality complements and not opening the platform to such an extent as to allow duplication of certain platform elements or unplanned competition to emerge.	(Anvaari & Jansen, 2010) (Boudreau, 2010)

	Feedback methods	The platform owner should consider implementing both developer and end-user feedback methods. This could possibly reduce information asymmetry (in cases where developers and platform owner compete, or competition between developers).	(Iansiti & Levien, 2004) (Scholten & Scholten, 2012)
	Programming languages	The programming languages used and required to develop complements on the platform is a decision the platform owner has to make as it has an effect on the adoption of the platform.	(Koch & Kerschbaum, 2014)
	Marketplaces	The marketplaces refer to the application portals where developers distribute their products and services and where end-users can purchase it. The platform owner should be aware of the different marketplace requirements.	(Koch & Kerschbaum, 2014)
	Data privacy and security	Depending on the target market, the platform owner needs to take extensive measures to ensure data privacy and security. For example, especially in health – related markets, extreme measures should to be taken to ensure privacy and security of data.	(Parker *et al*., 2016)
	Providers	Depending on the nature of the platform, it might have to be able to run on a specific hardware device (even other software (middleware), depending on type of platform) and the platform owner should ensure interfacing is possible.	(Van Alstyne *et al*., 2016)
Operational	Marketing & Sales	An effective marketing of the platform as well as sales approach can increase the growth of the ecosystem.	(Berk, Jansen, Luinenburg, *et al*., 2010) (Jansen *et al*., 2012)

	Research & Development	Especially in the IT industry, the rate of new technologies and concepts require constant R&D for new, innovative concepts to ensure that the ecosystem evolves. This can also be done in correspondence with the developer firms.	(Berk, Jansen, Luinenburg, *et al*., 2010) (Jansen *et al*., 2012)
	Support & Services	Adequate support and services should be available to help developers with initial set-up as well as other issues regarding the platform.	(Berk, Jansen, Luinenburg, *et al*., 2010) (Jansen *et al*., 2012)
	Market management	Market management forms a foundation of the competition within the ecosystem and the platform owner should create and manage trust, define the market governance structure, design a flexible market and leverage the participants in the ecosystem.	(Gawer & Cusumano, 2002)
	Reputation management	The reputation of the firm directly influences the attraction of stakeholder and ecosystem participants.	(Berk, Jansen & Luinenburg, 2010) (Van Angeren *et al*., 2016) (Gawer & Cusumano, 2014)
	Risk management	Risks can be shared amongst all partners as a method of risk management. This might include monitoring of potential risks, deciding to widen the ecosystem scope or monetary investments to evolve the technology.	(Parker & Van Alstyne, 2014b) (Gawer & Cusumano, 2014)
	Securing	Securing is the continuous process of modifying the level of control of the platform and can be related to managing the potential risks the platform and other actors may face.	(Ghazawneh & Henfridsson, 2013)

	Resourcing	Resourcing is the continuous process whereby the scope and diversity of the platform are increased enabling new resources, knowledge and capabilities within the ecosystem.	(Ghazawneh & Henfridsson, 2013)
	Investments	The firm’s revenue model should include the re-investment of profits into both the platform and the complements	(Jansen, 2014) (Gawer & Cusumano, 2014)
Key questions	How is the success of the platform defined and measured?		
	Is the platform an internal or external platform?		
	How will the responsibility and decision making within the ecosystem be divided?		
	How is the platform ecosystem defined?		

Developer			
Category	Concept	Description	Key reference(s) in formulating concept
Entry barriers	Homing costs	These are the costs associated with joining the ecosystem. It could include platform adoption, operating and opportunity costs such as access fees for access to the platform and its interfaces.	(Tiwana *et al*., 2010)
	Stickiness/exit barriers	These aspects define how hard it is for developers to leave the ecosystem.	(Jansen & Cusumano, 2012)
	Fairness	Fairness is a key attribute contributing to the sustainable relationships within the ecosystem. Fairness can also be related to justice which implies fair dealings within the ecosystem whether it be monetary or proprietary.	(Ryu *et al*., 2014) (Parker *et al*., 2016)
	Market size	The market size refers to the "footprint" of the platform ecosystem including the number of users within the ecosystem and the potential market that a developer can reach when joining the ecosystem.	(Koch & Kerschbaum, 2014) (Ryu *et al*., 2014) (Jansen, 2014)
	Marketplaces	Marketplaces are the application portals where developers upload their products and services and from where the end-user can obtain these. Depending on the platform nature, the platform might have to define which marketplaces their platform can comply with.	(Koch & Kerschbaum) (Jansen, 2014)

	Diversity	Diversity refers to both the variety in developers as well as the variety of the type of apps developed within the platform ecosystem and is a key component in a healthy, evolving ecosystem.	(Den Hartigh *et al*., 2006)
	Value creation	This refers to the extent that the platform enables the actors within the ecosystem to co-create and share value. The value creation opportunities within the ecosystem is a vital consideration for external developers as it reflects their potential success as a participant within the ecosystem.	(Rong *et al*., 2013) (Scholten & Scholten, 2012)
	Value distribution	This is the term given to describe in what manner the platform owner aims to divide the value (profits) throughout the ecosystem. An example would be whether the platform owner takes a percentage of developer profits, or if it is expected of the developer to pay a fixed amount that is not dependent on developer profits.	(Ghazawneh & Henfridsson, 2013)
	Trust	A key aspect in management and sustainability of platforms is establishing trust with developers. The platform owner often acts on behalf of the ecosystem and therefore to an extent carries the fate of the developers	(Cusumano & Gawer, 2002)
	Reputation	The platform leader should have a reputation of not stepping out of their product or services scope boundaries into the territory of the developers.	(Cusumano & Gawer, 2002)
	Credibility	A platform firm should invest in its brand credibility (believability) which in simple terms means that the platform owner does what it sets out to do. This can be directly linked to the loyalty of	(Ryu *et al*., 2014)

		developers to the platform.	
	Loyalty	Within the “ecosystem war”, platform owners should invest in encouraging loyalty of the developers to remain within their platform ecosystem.	(Ryu *et al*., 2014)
	Accessibility	The platform accessibility is related to the openness of the platform. This refers to the access that developers have to different levels of the platform and will be a key determining factor on whether to join a certain platform ecosystem or not.	(Anvaari & Jansen, 2010)
	Openness	The openness of the platform can be related not only to the architecture but also to the perceptions of the platform in terms of reputation, transparency and rights and is therefore a key aspect developers consider when deciding to join the platform.	(Van Angeren *et al*., 2016)
	Programming languages	The programming language(s) required for using the platform is a consideration when joining an ecosystem. It may require the developers to learn a new language or they can simply apply the languages they are familiar with.	(Koch & Kerschbaum, 2014) (Ryu *et al*., 2014)
	Toolkit quality	The toolkit (e.g. SDK) quality is one of the most important aspects in determining the loyalty of developers to a platform as it is linked to the quality of products and services they can deliver.	(Ryu *et al*., 2014) (Koch & Kerschbaum, 2014)
	Developer type	Mobile apps can either be native (run on OS and adapted for different devices), web-based (accessed via web-browser) or hybrid (web-apps in native browser). It should also be established whether	(Joorabchi *et al*., 2013)

		the platform owner scope allows for enterprise apps, commercial apps or apps for personal use. Basically the core proposition of each complementary firm.	
Governance structure	Intellectual Property	The platform leader should help the developers to protect their IP and should put their own interests aside for that of the ecosystem.	(Gawer & Cusumano, 2002) (Gawer & Cusumano, 2014)
	Data privacy and security	The personal information of the developers should be protected and the platform owner has to ensure the developers can secure the data of their products and/or services.	(Parker *et al*., 2016)
	Control mechanisms	Control mechanisms put in place by the platform owner which should encourage desirable behaviors in the developers. These could include market control, restrictive control, motivational control (funding, support systems), co-regulative control (development guidelines, rules and tools).	(Tiwana *et al*., 2010) (Scholten & Scholten, 2012)
	Design rules	These are the rules that the developers need to obey when using the platform to develop their own products and services. These rules need to be stable (all developers face same conditions) yet versatile (developers should still be able to innovate). The platform owner could possibly provide developer training for example.	(Tiwana *et al*., 2010)
	Goal congruency	The compatibility of the goals of the developer firms with that of the platform owner and ecosystem should be encouraged to reduce future tensions, competition as well as encourage innovations within the ecosystem.	(Scholten & Scholten, 2012)

Ecosystem	Tensions	Tensions between platform owner and complements. These tensions can occur when risks have to be taken within the ecosystem, when the objectives of the ecosystem participants differ or when competition occurs within the ecosystem.	(Tiwana *et al*., 2010) (Gawer & Cusumano, 2002)
	Interest of partners considered	The platform owner should consider the interests of all partners within the ecosystem, not only their own.	(Parker & Van Alstyne, 2014b)
	Managing network effects	The basic principle of network effects describes that the more developers join the ecosystem and develop complementary products and services using the platform, the more valuable the platform becomes and will therefore attract more complementary firms. Therefore the platform owner should be aware of the dynamics of these network effects and should generate methods of encouraging platform network effects and be vigilant for negative network effects.	(Olsson & Börjesson, 2009) (Scholten & Scholten, 2012)
	Encourage innovation	The platform owner should actively encourage innovation within the ecosystem, especially from the developers as a source of user innovation.	(Van Angeren *et al*., 2016) (Scholten & Scholten, 2012)
	Co-evolution	The platform owner should focus on co-evolving with the other actors in its ecosystem. This includes aspects such as re-evaluating ecosystem goals, platform markets etcetera)	(Den Hartigh *et al*., 2006)
Architecture	Interfacing	The ability to interface with other components or systems. In other words being compatible. If the platform is compatible, it can	(Ryu *et al*., 2014)

		interface with different platforms. An example is a document generating app being able to interface with Dropbox.	
	Feedback	The platform owner should enable the developers to provide feedback regarding the use of their platform. This forms a key part in the evolution and the continuous improvement of the platform.	(Jansen *et al*., 2012)
	Hardware and software integration	Different hardware components might require specific adaptations in software (for example screen resolution of different mobile phones). The platform owner should state whether development is possible for a variety of devices or not.	(Koch & Kerschbaum, 2014)
	Marketplace requirements	In order to enable the developers to develop better quality apps, the platform owner should be aware of the marketplace requirements.	(Koch & Kerschbaum, 2014)
	Leveraging	The firm should also leverage the ecosystem capabilities to add to scalability and adaptability.	(Gawer & Cusumano, 2002)
	Poor developer practice	This refers to aspects such as reduction in testing procedures of apps driven by developer laziness. Developer training regarding platform use might be solution to reduce this concern.	(Mansfield-Devine, 2012)
	Vulnerability	Possible weak points in the software should be eliminated, especially when working with sensitive data.	(Mansfield-Devine, 2012)
Standard	App development	Platform owners should be aware of the tools that major firms such as Apple (Apple HIG) and Google (Android) provide for guiding the	General

s & Models	guidelines	app-development process.	
	Partnership model		(Van Angeren *et al*., 2016) (Jansen & Cusumano, 2012)
	Boundary resources model		(Ghazawneh & Henfridsson, 2013)
	Open software enterprise model		(Jansen *et al*., 2012)
	SECO-SAM (Software ecosystem assessment model)		(Berk, Jansen, Luinenburg, *et al*., 2010)
Customer support	Debugging aids	Debugging aids should be included in the tools provided to the app developers.	(Ryu *et al*., 2014)
	Testing support	Testing applications in terms of mobility, network availability, sensors, etcetera are a major challenge for developers. Possible platform supported tools can be provided or recommend such as existing emulators (mimicking of hardware and software environments) and simulators (software environment).	(Joorabchi *et al*., 2013)
	Online communities	Connectedness amongst the developers as “niche players” in the ecosystem should be encouraged. This could result in community building within the ecosystem and could potentially increase innovation.	(Berk, Jansen, Luinenburg, *et al*., 2010)

	XMTs	Cross development tools (XMTs) are used to create apps for different operating systems using the same code.	(Koch & Kerschbaum, 2014)
	Ability to innovate & share	By sharing knowledge within the ecosystem, it can encourage innovation and lead to higher productivity. The level of control within the ecosystem is therefore a balance as control is needed, but the platform owner should not restrict the complementary firms to such an extent as to limit their ability to innovate.	(Berk, Jansen, Luinenburg, *et al*., 2010)
	Developer satisfaction	Developer satisfaction is key to ensure loyalty to the platform. These aspects also include ”soft” aspects such as the process being fun, intellectually stimulating, improving skills etcetera.	(Koch & Kerschbaum, 2014) (Ryu *et al*., 2014)
	Convenience of migration	Effort should be taken to reduce time, effort and budget requirement for developers migrating from different platforms and ecosystems.	(Joorabchi *et al*., 2013)
	Internal customer support	The platform owner should have a dedicated team to supply the platform users with the required support (support teams for example) regarding the use and debugging of the software platform.	(Jansen *et al*., 2012)
Control	Monitoring	The platform firm should monitor all actors in the ecosystem, evaluating their performance, making decisions and taking actions based on the observations.	(Lima *et al*., 2016)
	Track user loyalty	The number of active developers and their loyalty could be tracked as this can be a key indication of a more attractive ecosystem	(Jansen, 2014)

		(competition).	
	Review process	The platform owner should carefully review the products and services that are developed on their platform as a method of quality control.	(Nayebi *et al*., 2012) (Ryu *et al*., 2014)
Key questions	How is success measured in terms of the developers using the platform?		
	What are the main challenges developers will face when using this platform?		
	Who are the other ecosystems competing to acquire developers?		
	What are the entry barriers to joining this ecosystem?		
	Who are the “type” of developers we want to attract to our ecosystem?		

End-user			
Category	Concept	Description	Key references
Context of use (important as the platform owner might also be developing apps (internal platform))	Organisational context	The organisational context should be considered as to ensure the work and safety rules and regulations are not violated with the use of the app.	(Seffah *et al*., 2006) - Obtained from ISO 9241-11 (1998)
	Physical context	The physical surroundings of the user such as possible noise, ambient conditions, health and safety issues etcetera.	(Seffah *et al*., 2006) - Obtained from ISO 9241-11 (1998) (Harrison *et al*., 2013)
	Social context	The social context refers to aspects such as whether assistance is available, is the envisioned environment for a single user or multi-users.	(Seffah *et al*., 2006) - Obtained from ISO 9241-11 (1998)
	Task characteristics	The task characteristics includes the frequency of use of the app and the corresponding duration, physical and mental demands and the app complexity.	(Seffah *et al*., 2006) - Obtained from ISO 9241-11 (1998) (Harrison *et al*., 2013)
	User characteristics	User characteristics is vital to be attentive to. It includes the languages, the computer/digital skills, the reading level, the level of experience etcetera, which will inherently determine the success of the app.	(Seffah *et al*., 2006) - Obtained from ISO 9241-11 (1998)

	Country differences	Firstly, app user behavior differs in different countries in terms of what type of apps users prefer, the features they dislike in apps or to what extent they give app feedback (see reference). Secondly, each country has different laws and regulations related to data, content being displayed, freedom of speech etcetera.	(Lim *et al*., 2015)
	Geographical context	The geographical context of app use refers to the network coverage available (2G, 3G, 4G, etc). This should be considered when designing the app as it will affect how the app is presented.	(Harrison *et al*., 2013)
Quality control (as an ecosystem perspective is undertaken, the products of developers affects the ecosystem)	Usability	Usability attributes and metrics form a large part of the literature regarding app development and use. Various standards and models include usability attributes such as efficiency, effectiveness, satisfaction, reliability, memorability and low error rate.	IEEE standards, ISO standards, Other literature models
	Learnability	The level of ease with which the app can be used to its fullest extent. This could for example be a determining factor if time is crucial to the customer.	(Seffah *et al*., 2006)
	Understandability	Understand ability includes aspects such as navigation through the app levels, logic of using the app, confusing text or button layout, user guidance, simplicity etcetera.	(Seffah *et al*., 2006)
	Reliability and performance	The platform owner should ensure on their part the smooth, error free operation of apps that are stable, reliable (especially	(Franko & Tirrell, 2012) (Seffah *et al*., 2006)

		in healthcare) and can meet the performance requirements of the developers and end-users.	
	App deployment	In the case of enterprise apps for example, it might be necessary to assist the end-user with guidelines for using the app.	General
	User requirements	The developer of the app should conduct an investigation to determine the user requirements of the proposed app as this will provide the ability to evaluate the success of the app – how well it meets the user needs and requirements.	(Lim *et al*., 2015)
Feedback	User data feedback	In order to continuously improve the platform, end-user feedback should be enabled. This can be done through mining marketplace data, incorporating activity logs, through surveys, customer feedback or controlled-environment experiments.	(Pagano & Maalej, 2013) (Nayebi *et al*., 2012)
	Quick updates	The software should enable the quick updating of apps if major issues are reported via feedback. Feedback could even lead to the platform software being updated.	General
Standards & Models	Laws and regulations	The platform owner may decide to invest in checking if the apps developed on their platform complies with the known applicable laws and regulations. If the developers do not comply, it may have a negative effect on the ecosystem as a whole and on the platform reputation. For example FDA laws in USA for the healthcare industry, or Appstore specific rules.	(Lim *et al*., 2015)

	ISO Standards	Or any standards related to usability or other context specific aspects should be obeyed.	General
	Apple HIG	Guidelines provided includes platform characteristics, human interface principles and user experience guidelines.	(Apple Inc., n.d.)
	Android UIG		(Android, n.d.)
	Usability Engineering	Usability engineering by Jakob Nielsen	(Nielsen, 1994)
	MUSiC	Metrics for Usability Standards in Computing	(Seffah *et al*., 2006)
	QUIM	Quality of Use Integrated Measurement model	(Seffah *et al*., 2006)
	PACMAD	People At The Centre of Mobile Application Development	(Harrison *et al*., 2013)
	GQM Approach	Goal Question Metric Approach (for software engineering)	(van Solingen, Basili, Caldiera & Rombach, 2002)
Attractiveness (first impression which is important for competitive edge)	Visual aspects	The aspects such as the app icon and the visual layout of the app interface are important aspects not only if selling the app in a marketplace, but even for enterprise apps as this is directly related the user experience of the app. This can also include aspects such as app colours and fonts.	(Lim *et al*., 2015) (Seffah *et al*., 2006)
	Level of exposure	This aspect can be related to network effects and refers to the exposure of the app in terms of the amount of other users. App users consider the popularity of the app when buying or using a	(Lim *et al*., 2015)

		specific app.	
	Pricing	Much thought needs to be placed on the pricing of the apps and to relate this to the end-user target market.	(Lim *et al*., 2015)
	User comments	The ability for app users’ to comment on aspects such as app crashing, it being too slow, difficulty of use, unnecessary features, etc.	(Poppinga & Henze, in press) (Lim *et al*., 2015)
	Other marketplace related factors	There are several literature guidelines presenting the strategies that should be followed regarding success in the marketplace (Appstore for example). This includes the app description, screenshots, designing of more appealing apps etcetera. The marketplace dynamics are outside the scope of this research.	(Lim *et al*., 2015)
	Pricing guidelines	The platform owner can possibly provide guidelines in terms of pricing of apps based on their platform. If the apps are too expensive, no one will buy it and therefore it having an indirect effect on the platform.	General
Key questions	How success is measured in terms of the end-users?		
	What can the platform owner do to encourage the usage of apps developed on their platform?		
	What are the main reasons for app abandonment (does it reflect the platform or developer)?		
	Is the app usable and does it meet the user requirements?		

REFERENCES

African Union Department of Social Affairs. 2017. African Health Stats. [Online], Available: http://www.africanhealthstats.org/ [2017, May 12].

Van Alstyne, M.W., Parker, G.G. & Choudary, S.P. 2016. Pipelines, Platforms, and the New Rules of Strategy. Harvard Business Review. 94(4):54–62. [Online], Available: http://library.capella.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bah&AN=113934414&site=ehost-live&group=alumni.

Android. n.d. User Interface Guidelines. [Online], Available: https://developer.android.com/guide/practices/ui_guidelines/index.html [2017, November 09].

Van Angeren, J., Alves, C. & Jansen, S. 2016. Can we ask you to collaborate? Analyzing app developer relationships in commercial platform ecosystems. Journal of Systems and Software. 113:430–445.

Anvaari, M. & Jansen, S. 2010. Evaluating Architectural Openness in Mobile Software Platforms. 4th European Conference of Software Architecture. (c):85–92.

Apple Inc. n.d. Human Interface Guidelines. [Online], Available: https://developer.apple.com/design/ [2017, November 09].

Assamala. 2014. African Regional Health Report 2014. World Health Organisation. 187.

Baars, A. & Jansen, S. 2012. A Framework for Software Ecosystem Governance. Icsob 2012.

Baldwin, C.Y. & Woodard, C.J. 2008. The Architecture of Platforms: A Unified View. Working Papers -- Harvard Business School Division of Research. 1–31.

Barbosa, O. & Alves, C. 2011. A Systematic Mapping Study on Software Ecosystems. Proceedings of the Workshop on Software Ecosystems 2011. 15–26.

Berk, I. Van Den, Jansen, S. & Luinenburg, L. 2010. Software Ecosystems : A Software Ecosystem Strategy Assessment Model. (c).

Berk, I. Van Den, Jansen, S., Luinenburg, L. & van den Berk, I. 2010. Software ecosystems. Proceedings of the Fourth European Conference on Software Architecture Companion Volume - ECSA ’10. (c):127.

Bhih, A.A., Johnson, P. & Randles, M. 2016. Diversity in Smartphone Usage. Proceedings of the 17th International Conference on Computer Systems and Technologies 2016 - CompSysTech ’16. 81–88.

Boudreau, K. 2010. Open Platform Strategies and Innovation: Granting Access vs. Devolving Control. Management Science. 56(10):1849–1872.

Cusumano, M.A. & Gawer, A. 2002. The Elements of Platform Leadership. MIT Sloan Management Review. 43(3).

Franko, O.I. & Tirrell, T.F. 2012. Smartphone app use among medical providers in ACGME training programs. Journal of Medical Systems. 36(5):3135–3139.

Gawer, A. 2009. Platforms, markets and innovation.

Gawer, A. 2014a. Bridging differing perspectives on technological platforms: Toward an integrative framework. Research Policy. 43(7):1239–1249.

Gawer, A. 2014b. Bridging differing perspectives on technological platforms: Toward an integrative framework. Research Policy. 43(7):1239–1249.

Gawer, A. & Cusumano, M. 2013. Industry Platform and Ecosystem Innovation. Journal of Product Innovation and Management. 31(3):417–433.

Gawer, A. & Cusumano, M.A. 2002. Platform Leadership: How Intel, Microsoft and Cisco Drive Industry Innovation. Boston, Massachusetts: Harvard Business School Press.

Gawer, A. & Cusumano, M.A. 2008. How Companies Become Platform Leaders. MIT Sloan Management Review. 49(2):28–35.

Gawer, A. & Cusumano, M.A. 2014. Industry platforms and ecosystem innovation. Journal of Product Innovation Management. 31(3):417–433.

Ghazawneh, A. & Henfridsson, O. 2013. Balancing platform control and external contribution in third-party development: The boundary resources model. Information Systems Journal. 23(2):173–192.

GSMA. 2017. The Mobile Economy 2017. 1–56.

Harman, M., Jia, Y. & Zhang, Y. 2012. App store mining and analysis: MSR for app stores. IEEE International Working Conference on Mining Software Repositories. 108–111.

Harrison, D. 2009. An Overview of Health and Health care in South Africa 1994 – 2010 : Priorities , Progress and Prospects for New Gains. A Discussion Document Commissioned by the Henry J Kaiser Family Foundation to Help Inform the National Health Leaders’ Retreat Muldersdrift. (December 2009):1–40. [Online], Available: http://www.doh.gov.za/docs/reports/2010/overview1994-2010.pdf.

Harrison, R., Flood, D. & Duce, D. 2013. Usability of Mobile Applications: Literature Review and Rationale for a New Usability Model. Journal of Interaction Science. 1(1):1–16.

Den Hartigh, E., Tol, M. & Visscher, W. 2006. The Health Measurement of a Business Ecosystem. ECCON 2006 Annual Meeting. 2783565(secretary 2781150):1–39. [Online], Available: http://www.chaosforum.com/docs/nieuws/health.pdf.

Harvey, M.J. & Harvey, M.G. 2014. Privacy and Security Issues for Mobile Health Platforms. 65(7):1305–1318.

van Heerden, A., Harris, D.M., van Rooyen, H., Barnabas, R. V., Ramanathan, N., Ngcobo, N., Mpiyakhe, Z. & Comulada, W.S. 2017. Perceived mHealth barriers and benefits for home-based HIV testing and counseling and other care: Qualitative findings from health officials, community health workers, and persons living with HIV in South Africa. Social Science & Medicine. 183:97–105.

Herman, H., Grobbelaar, S.S. & Pistorius, C.W.I. 2017. Exploring Key Concepts in the Technology Platform Literature: A Systematic Review. In Proc. 28th Conference of South African Institute for Industrial Engineers (SAIIE).

Iansiti, M. & Levien, R. 2004. The Keystone Advantage. Boston, Massachusetts, USA: Harvard Business School Press.

Jabareen, Y. 2009. Building a conceptual framework: philosophy, definitions, and procedure. International Journal of Qualitative Methods. 8:49–62.

Jansen, S. 2014. Measuring the health of open source software ecosystems: Beyond the scope of project health. Information and Software Technology. 56(11):1508–1519.

Jansen, S. & Cusumano, M. 2012. Defining software ecosystems: A survey of software platforms and business network governance. CEUR Workshop Proceedings. 879:41–58.

Jansen, S., Finkelstein, A. & Brinkkemper, S. 2009. A sense of community: A research agenda for software ecosystems. 31st International Conference on Software Engineering - Companion Volume. 187–190.

Jansen, S., Brinkkemper, S. & Finkelstein, A. 2009. Business network management as a survival strategy: A tale of two software ecosystems. Proceedings of the First Workshop on Software Ecosystems. (2):34–48.

Jansen, S., Brinkkemper, S., Souer, J. & Luinenburg, L. 2012. Shades of gray: Opening up a software producing organization with the open software enterprise model. Journal of Systems and Software. 85(7):1495–1510.

Joorabchi, M.E., Mesbah, A. & Kruchten, P. 2013. Real challenges in mobile app development. International Symposium on Empirical Software Engineering and Measurement. 15–24.

Kahn, J.G., Yang, J.S. & Kahn, J.S. 2010. “Mobile” Health Needs And Opportunities In Developing Countries. 2(2):252–258.

Koch, S. & Kerschbaum, M. 2014. Joining a smartphone ecosystem: Application developers’ motivations and decision criteria. Information and Software Technology. 56(11):1423–1435.

Lim, S.L., Bentley, P.J., Kanakam, N., Ishikawa, F. & Honiden, S. 2015. Investigating country differences in mobile app user behavior and challenges for software engineering. IEEE Transactions on Software Engineering. 41(1):40–64.

Lima, T., Santos, R.P. dos, Oliveira, J. & Werner, C. 2016. The importance of socio-technical resources for software ecosystems management. Journal of Innovation in Digital Ecosystems. 3(2):98–113.

Manikas, K. & Hansen, K.M. 2013. Reviewing the health of software ecosystems - A conceptual framework proposal. CEUR Workshop Proceedings. 987:26–37.

Mansfield-Devine, S. 2012. Android architecture: Attacking the weak points. Network Security. 2012(10):5–12.

Moazed, A. 2016. What is a Platform? [Online], Available: https://www.applicoinc.com/blog/what-is-a-platform-business-model/ [2017, October 19].

Moore, J.F. 1993. Predators and prey: a new ecology of competition. Harvard Business Review. 71(3):75–86.

Nayebi, F., Desharnais, J.-M. & Abran, A. 2012. The state of the art of mobile application usability evaluation. 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). (May):1–4.

Nielsen, J. 1994. Usability Engineering. 1st ed. Morgan Kaufmann.

Oh, D.S., Phillips, F., Park, S. & Lee, E. 2016. Innovation ecosystems: A critical examination. Technovation. 54(March):1–6.

Olsson, U. & Börjesson, M. 2009. Leveraging open source in commercial service layer development - A case study. 2009 13th International Conference on Intelligence in Next Generation Networks: Beyond the Bit Pipes, ICIN 2009.

Pagano, D. & Maalej, W. 2013. User feedback in the appstore: An empirical study. 2013 21st IEEE International Requirements Engineering Conference, RE 2013 - Proceedings. 125–134.

Parker, G. & Van Alstyne, M. 2013. Innovation, Openness & Platform Control. Proceedings of the 11th ACM conference on Electronic commerce - EC ’10.

Parker, G. & Van Alstyne, M.W. 2014a. Platform Strategy. SSRN Electronic Journal. 1–14.

Parker, G. & Van Alstyne, M.W. 2014b. Platform Strategy. SSRN Electronic Journal. (1967):1–14.

Parker, G.G., Van Alstyne, M.W. & Choudary, S.P. 2016. Platform Revolution: How Networked Markets Are Transforming the Economy-And How to Make Them Work for You. W. W. Norton & Company.

Peltoniemi, M. 2006. Preliminary theoretical framework for the study of business ecosystems. Emergence: Complexity & Organization. 8(1):10–19.

Poppinga, B. & Henze, N. (in press). My App is an Experiment : Experience from User Studies in Mobile App Stores. International Journal of Mobile Human Computer Interaction. Volume 3 I:Pages 71-91.

Rong, K., Lin, Y., Shi, Y. & Yu, J. 2013. Linking business ecosystem lifecycle with platform strategy: A triple view of technology, application and organisation. International Journal of Technology Management. 62(1):75–94.

Ryu, M.H., Kim, J. & Kim, S. 2014. Factors affecting application developers’ loyalty to mobile platforms. Computers in Human Behavior. 40:78–85.

Scholten, S. & Scholten, U. 2012. Platform-based Innovation Management: Directing External Innovational Efforts in Platform Ecosystems. Journal of the Knowledge Economy. 3(2):164–184.

Seffah, A., Donyaee, M., Kline, R.B. & Padda, H.K. 2006. Usability measurement and metrics: A consolidated model. Software Quality Journal. 14(2):159–178.

van Solingen, R., Basili, V., Caldiera, G. & Rombach, H.D. 2002. Goal Question Metric (GQM) Approach. Encyclopedia of Software Engineering. 2:1–10.

Tiwana, A., Konsynski, B. & Bush, A.A. 2010. Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. Information Systems Research. 21(4):675–687.

Wasserman, A.I. & Fosser. 2010. Software Engineering Issues for Mobile Application Development. FoSER ’10 Proceedings of the FSE/SDP workshop on Future of software engineering research. 397–400.

Zapata, B.C., Fernández-Alemán, J.L., Idri, A. & Toval, A. 2015. Empirical Studies on Usability of mHealth Apps: A Systematic Literature Review. Journal of Medical Systems. 39(2):1–19.