DESIGNING BUSINESS MODELS FOR PLATFORM AS A SERVICE: TOWARDS A DESIGN THEORY

Research-in-Progress

Andrea Giessmann
University of Lausanne, Faculty of Business and Economics (HEC)
CH-1015 Lausanne, Switzerland
andrea.giessmann@unil.ch

Christine Legner
University of Lausanne, Faculty of Business and Economics (HEC)
CH-1015 Lausanne, Switzerland
christine.legner@unil.ch

Abstract

Platform as a Service (PaaS) solutions are changing the ways that software is produced, distributed, consumed, and priced. Unlike Software as a Service (SaaS) or Infrastructure as a Service (IaaS), PaaS allows for value co-creation by offering complementary components and applications that are developed in emerging ecosystems of third party developers. Despite increasing interest among practitioners and researchers, there has been little work in understanding how PaaS business models should be designed to establish a flourishing ecosystem. We seek to develop a design theory that facilitates the design of PaaS business models, taking into account the specifics of multisided business models. Four meta-requirements describe the purpose and scope of our design theory, and six design principles guide PaaS providers in designing effective business models. By focusing on designing business models, our research goes beyond previous approaches for studying PaaS.

Keywords: Business model, cloud computing, platform design, platform ecosystem

Introduction

Platform as a Service (PaaS) solutions are changing the ways that software is produced, distributed, consumed, and priced. Unlike Software as a Service (SaaS) or Infrastructure as a Service (IaaS), PaaS allows for value co-creation by offering complementary components and applications that are developed in emerging ecosystems of third party developers (Tiwana et al. 2010). "A burgeoning body of research has started to theorize about how such ecosystems are formed and their implications for platform owners, complementary providers, and users" (Ceccagnoli et al. 2012). However, despite increasing interest among practitioners and researchers, there has been little work in understanding how PaaS business models should be designed to establish a flourishing ecosystem around these platforms. This is a significant gap in understanding. This paper at seeks to develop a design theory that facilitates the design of PaaS business models, taking into account the specifics of multisided business models. Thus, the research question is as follows: What are relevant design goals and design principles for PaaS business models?

Based on Gregor's taxonomy of theory, the goal is to develop a type five theory for design and action (Gregor 2006). This type of theory lays out how to do something and "gives explicit prescriptions [...] for constructing an artifact" (Gregor 2006). Gregor and Jones (2007) propose eight components to document a design theory. However, since our research is still in progress, we concentrate on the following five components: (1) the purpose and scope of our theory is described by four meta-requirements, before we define (2) the main construct our theory is based on. The main contribution will be (3) a set of design principles supported by (4) justificatory knowledge that guides the effective design of PaaS business models. An (5) expository instantiation serves for theory exposition. The study utilizes action design research (ADR), which – according to (Sein et al. 2011) – is a "research method for generating prescriptive design knowledge through building and evaluating ensemble IT artifacts in an organizational setting." Our research's organizational context is the new PaaS solution (here called SHC) of one of the largest global software companies.

Prior Research

Platforms in the software industry are defined as "...the extensible codebase of a software-based system that provides core functionality shared by the modules that interoperate with it and the interfaces through which they interoperate" (Tiwana et al. 2010). Platform as a Service refers to software platforms that have primarily been discussed in the context of cloud computing. According to the most cited architectural concepts of cloud computing, PaaS represents the middle layer, connecting the Infrastructure as a Service (IaaS) and Software as a Service (SaaS) cloud layers (Höfer and Karagiannis 2011; Marston et al. 2011; Mell and Grance 2011; Vaquero et al. 2009; Zhang et al. 2010). Based on a systematic literature review, Giessmann and Stanoevska (2012) describe PaaS as an execution environment in which external developers deploy and run their complementary components and applications. PaaS facilitates the development, testing, and management of software components, as well as knowledge exchange between developers.

The current PaaS market is a fast-growing but largely fragmented market (IDC 2012). It is expected that there will be a market consolidation towards only a few large PaaS providers offering a comprehensive PaaS suite (Forrester 2011; Gartner 2012). But, "who wins and who loses these competitions is not simply a matter of who has the best technology or the first product. It is often who has the best platform strategy and the best ecosystem to back it up" (Cusumano 2010). Although the PaaS phenomenon only emerged recently and is associated with cloud computing, it shows the characteristics of a multisided platform. The latter coordinates distinct groups of customers who need each other in some way (Evans 2003) and provides infrastructure and rules that facilitate the two groups' transactions (Eisenmann et al. 2006). Evans (2003) divides multisided platforms into three categories: market-makers, audience-makers, and demand coordinators. PaaS solutions can be assigned to market-maker platforms that "enable members of distinct groups to transact with each other. Each member of a group values the service more highly if there are more members of the other group – because that increases the likelihood of a match and reduces the time it takes to find an acceptable match" (Evans 2003). Hence, to succeed, PaaS providers must get and keep on board two or more distinct customer groups.

Cusumano and Gawer (2002) have developed four levers of platform leadership, to assist managers in strategy formulation and implementation: 1) Scope, which is a company's amount of internal innovation

and how much it encourages outsiders to do. 2) Product technology refers to decisions on the architecture, for instance, what functionality or features to include in the platform or how open its interface should be. 3) External relationships with complementors is the process by which the platform leader manages complementors and encourages them to contribute to a vibrant ecosystem. 4) Internal organization is the right internal structure, which helps platform producers manage external and internal conflicts of interest (Cusumano and Gawer 2002; Gawer and Cusumano 2008). While these four levers are well recognized in strategic management literature, they provide little guidance for the design of PaaS business models.

Eisenmann et al. (2006) note that multisided platform business models differ fundamentally from other offerings' business models. "In the traditional value chain, value moves from left to right: To the left of the company is cost; to the right is revenue. In multi-sided business models, cost and revenue are both to the left and the right, because the platform has a distinct group of users on each side." They identify three key challenges for multisided platform providers: pricing the platform, winner-takes-all dynamics, and the threat of envelopment. The authors focus on providing guidance for executives on how these challenges should be considered in designing a platform's business model. However, in line with Rochet and Tirole (2003) as well as Parker and Van Alstyne (2008), Eisenmann et al. (2006) believe that "the key decision here is pricing." At present there is a lack of systematic research about the specific design of multisided business models for PaaS (Giessmann and Stanoevska 2012). Specifically, we find that prior research on multisided platforms and associated concepts such as network effects may provide a suitable theoretical lens to study business models for PaaS.

Methodology

Our research will seek to develop a design theory – in line with Gregor and Jones (2007) – that facilitates the design of PaaS business models. Thus, we employ the design science research paradigm (Hevner et al. 2004; March and Smith 1995). Specifically, our design theorizing contains goal-oriented prescriptions on designing business models for PaaS solutions. We utilized action design research (ADR) – according to Sein et al. (2011) – as a "research method for generating prescriptive design knowledge through building and evaluating ensemble IT artifacts in an organizational setting."

Our ADR approach covers all four stages of the process (see Table 1) and has been accomplished in close collaboration with a large enterprise software corporation (here called Alpha[1]). Alpha is one of the largest global software companies and offers enterprise resource planning systems as well as enterprise data warehouse products as its primary products. With its PaaS solution, SHC, Alpha offers a powerful Java-based platform that provides sophisticated development and integration capabilities. However, Alpha is struggling with its PaaS solution, for different reasons: SHC has fallen short of expectations and has not met several assumptions from the business case. In particular, development costs have been higher than planned, and Alpha failed to meet the planed sales figures by far. Traditionally, Alpha has developed and distributed license-based software packages. Offering a software platform where external developers deploy and run their complementary components is changing their way of doing business. It was found that an innovative and effective business model was needed for this new kind of software platform. This was the trigger for our research, which addresses a practice-inspired research problem as well as the following class of problems: How should a business model for PaaS solutions be designed?

The PaaS business model design follows an organization-dominant building, intervention, and evaluation (BIE) schema, since design knowledge will be generated where the primary source of innovation is organizational intervention. In the first BIE cycle, the current business model of Alpha's SHC solution was analyzed, including trigger analyses and investigations of customer needs. A first version of the artifact, in the form of SHC's status quo business model, was developed by using the business model framework by Johnson et al. (2008). As part of the first reflection and learning cycle, 23 PaaS business models have been investigated. Key insights of this exercise went into the second BIE cycle.

Using several methods from the business model innovation (BMI) area, including blue ocean strategy (Kim and Mauborgne 2004), kill the company (Bodell 2012), and BMI pattern cards (Gassmann et al. 2013), more than 200 BMI ideas were created during workshops to improve the SHC business model. The project

[1] Company's name and solution are blinded.

team clustered and prioritized these ideas and developed nine BMI options in detail. These nine BMI options were evaluated with 13 experts outside the ADR team. The interviewed experts had the following positions/roles: business development cloud, product management, solution management, chief product owner, ecosystem and channels (2x), business development, sales head APJ, custom development, North American cloud sales, sales management, and head of cloud integration. The evaluation comprised two iterations – a qualitative interview on expert opinions per BMI option and a quantitative evaluation using the following evaluation criteria: revenue potential, customer acceptance, impact on critical mass, differentiation/thought leadership, costs, risks, conflicts, and required time. Finally, based on a management presentation, Alpha decided to implement three of the nine proposed BMI options for its SHC solution. Thus, as a result of the second BIE cycle, an updated version of the SHC business model was created.

We consulted all materials such as minutes from the 19 meetings, four one-day workshops, and 13 expert interviews, documentation, slides, reports, and results; we analyzed these within the reflection and learning cycles conducted in parallel. In addition, an exhaustive market analysis was conducted, and literature on platform strategies, network effects, critical mass, and business model research was reviewed. In the formalization of the learning stage, we sought to convert our situated learning into components of a design theory based on Gregor and Jones (2007). We synthesized meta-requirements, as well as principles of form and function that we developed and formulated in full during the BIE cycles.

Table 1. Action Design Research (ADR) Process, in line with (Sein et al. 2011)

Stages and principles		**Artifact**
Stage 1: Problem formulation		
Principle 1: Practice-inspired research	Our research was driven by software providers' practical need to offer innovative and effective business models for PaaS.	**Recognition:** The PaaS solution has fallen short of expectations and has not met several assumptions from the business case. Competitors are much more successful in the market. Recognition that an innovative and effective business model is needed.
Principle 2: Theory-ingrained artifact	The artifact created and evaluated via ADR will be a business model for PaaS, informed by established business model theories such as those by Johnson et al. (2008), Osterwalder and Pigneur (2010), Timmers (1998), and Morris et al. (2005). PaaS shows all the characteristics of multisided platforms, and the associated theories and concepts (e.g., network effects) are also taken into account.	
Stage 2: Building, intervention, and evaluation (BIE)		
Principle 3: Reciprocal shaping	A trigger analysis – investigating internal and external opportunities and threats – has been performed, as well as an investigation of customer needs. Problems encountered were iteratively addressed and formulated as early design principles in collaboration with practitioners.	**First BIE cycle:** The artifact – a business model representation of SHC's business model – documented the status quo, thus making transparent the identified weak points. **Second BIE cycle:** 9 different instances of business models were created for the SHC solution. Finally, the winning 3 BMI options were integrated into a final, updated business model.
Principle 4: Mutually influential roles	The ADR core team included 1 moderator, 2 researchers, 2 solution managers, 1 sales representative, 1 finance representative, and 1 platform architect, in order to include theoretical, technical, and practical perspectives. The lead designer was SHC's head of solutions management.	
Principle 5: Authentic and concurrent evaluation	The 9 identified BMI options were evaluated with a total of 13 experts outside the ADR core team in two iterations: first from a qualitative and second from a quantitative perspective. The qualitative evaluation mainly included open questions that also led to refinements of the proposed BMI options. The quantitative evaluations were based on an evaluation framework developed by the ADR team. Experts had to rate the BMI options among different criteria by using a scoring model.	
Stage 3: Reflection and learning		
Principle 6: Guided emergence	First, an investigation of 23 PaaS competitors' business models was performed, to gain a deep understanding of the PaaS market. Second, a literature review was conducted on platform strategies, business model design as well as innovation, network effects, and critical mass. The collected and categorized qualitative data, representing the situated learning, was then transferred to the broader class of problems: designing business models for PaaS.	**Emerging version and realization:** New meta-requirements for the business model artifact based on results that emerged in the BIE cycles. A revised version of the initial design principles.

Stage 4: Formalization of learning		
Principle 7: Generalized outcomes	A set of design goals and principles for PaaS business models was articulated, positioning Alpha's business model for SHC as an instance.	**Ensemble version:** An ensemble embodying the design goals and principles.

Designing Business Models for Platform as a Service

Purpose and scope

As noted, PaaS solutions are multisided platforms and therefore require multisided business models. Consequently, the first meta-requirement addresses the key challenge to *get and keep on board all relevant customer segments* (MR1). "Multi-sided platforms coordinate the demand of distinct groups of customers who need each other in some way" (Evans 2003). Transferred to PaaS, this means the solution must coordinate consumer needs for complementary components and applications, as well as external developers' demand for a large installed customer base (Ceccagnoli et al. 2012). Hence, MR2 seeks to *achieve a critical mass of complementary components and applications as well as consumers*. For consumers, the value of components and applications on the platforms increases with the expected number of components and applications available on the platform (Economides 1996). For third party developers, the platform's value increases with an increasing number of users in the installed base. Thus, the third meta-requirement seeks to *leverage positive network effects* (MR3). Like any other business, PaaS solution providers seek to earn profit. However, they also face the challenge of protecting their "sources of profit while enabling complementors to make an adequate profit" (Gawer and Cusumano 2008). Accordingly, we observed that third party developers refuse to join platforms if they must first sign expensive license contracts. The PaaS market is still emerging. PaaS providers not only need to invest in platform development, but also have "to create a strong ecosystem of developers and consumers around the own platform" (Giessmann and Stanoevska 2012). Maximizing "short-term profits [..] may not encourage a global ecosystem of complementors to develop over the long term" (Gawer and Cusumano 2008). MR4 therefore addresses the *mid-term to long-term profitability* of PaaS business models.

Constructs

Since our theory's material artifact is a business model for PaaS solutions, the main construct our theory is based on is business models. In recent years, several studies – such as Ballon (2007); Chesbrough (2007); Johnson et al. (2008); Mahadevan (2000); Morris et al. (2005); Osterwalder et al. (2005); Timmers (1998); Zott et al. (2011) – have noted the importance of actively analyzing and designing business models. As noted, PaaS business models need to get and keep on board two or more distinct customer groups. To consider different customer segments and their needs, we applied the business model framework of Johnson et al. (2008), according to whom a business model consists of four interlocking elements – customer value proposition, profit formula, key resources, and key processes – that, taken together, create and deliver value. Besides the definition of business models, Pateli and Giaglis (2004) distinguish seven additional business model research areas: components/fundamental constructs, taxonomies used for categorizations of business models, conceptual models, design methods and tools, adoption factors, evaluation models, and change methodologies. We aim to develop a design theory for PaaS business models. Hence, we focus on the fourth research field: design methods and tools, that is, "building methods and developing tools for designing business models" (Pateli and Giaglis 2004).

Principles of form and function

The first design principle (DP1) addresses the identification of customer segments. As noted, a PaaS business model must identify at least two or more distinct customer segments. From our analysis of PaaS business models, we find that PaaS providers have the options to address four possible customer groups: 1) consumers of components and applications, 2) development partners, including independent software vendors (ISVs) as well as system integrators (SIs) such as IT consultancies, 3) individual developers that are not yet building commercial solutions (such as students or startups), and 4) platform customers as a possible customer group, that is, enterprises using PaaS in the sense of a private cloud (Stanoevska-Slabeva and Wozniak 2009) where they develop and run components and applications for in-house utilization.

In line with Johnson et al. (2008), we believe the most important aspect to get right is the value proposition, that is, the main services a PaaS provider offers its customer segments. Based on our studies, four types of platforms were derived that describe the most important value propositions: First, PaaS's main value proposition can be to facilitate the development of components and applications (development-focused platforms). Second, an alternative value proposition for PaaS can be the integration of the developed applications into an existing SaaS solution (application-based integration). The third group of PaaS providers offers a distribution channel as its most important service (e.g., Facebook developers). The fourth group seeks to integrate any combination of on-premise and on-demand applications. Hence, DP2 defines a core value proposition, that is, the most important services and features for each of the identified customer segments.

DP3 relates to the insight that a platform without content (i.e. components and applications) is not attractive for all identified customer segments and is therefore unlikely to achieve critical mass. Developing customers rely on existing components in order to be able to work efficiently with the platform, and consumers prefer a variety of applications. Consequently, DP3 implies starting with an attractive of set of components and applications. There are several options to achieve this, and they can be combined: PaaS providers develop an initial set of components and applications themselves or hire third party developers to do this. Additionally, the developing side of the business model can be subsidized (Ceccagnoli et al. 2012; Eisenmann et al. 2006; Evans 2003; Gawer and Cusumano 2007), or the PaaS provider might invest in the developing side by offering developer conferences, summits, and competitions. PaaS providers need to announce their investments into the platform and to keep entry barriers low by supporting established languages and development tools (Evans 2003).

Another key insight of our research is that PaaS providers need to continuously take care of the needs and demands of their installed base, or, in the words of Gawer and Cusumano (2007), to "keep innovating on the core, ensuring that it continues to provide an essential (and difficult to replace) function to the overall system." DP4 therefore advises leveraging existing customers of the platform. Ways to achieve this include custom development and maintenance, integration of customers into roadmap planning, and offering matchmaking capabilities, where consumers are linked to developers and vice versa. Most importantly, however, PaaS providers must ensure variety and quality in the components and applications offered on the platform.

Successful PaaS solutions actively "manage relationships with complementors" (Gawer and Cusumano 2007), and DP5 advises addressing this explicitly in the business model design. The design can form a continuum, with several possible points along the way. One end of the continuum is the granting of exclusive rights to complementors, i.e. the PaaS provider does not authorize competing components and applications in order to protect a complementor. At the other end of the continuum are platforms that refuse to give any commitments to complementors, but actively search and imitate or buy promising components and applications. Intermediate steps include the granting of intellectual property rights (IPR) or copyrights, as well as the integration of complementors into roadmap planning of PaaS solutions.

In line with Cusumano and Gawer (2002), we are of the view that that designing business models for PaaS requires designing the right internal structure. Cusumano and Gawer (2002) identified three design options: 1) keeping groups with similar goals under one manager, 2) addressing organizational culture and processes, and 3) improving internal communication of corporate strategy. In addition, our research revealed two more options that might not be applicable to every PaaS: 4) Providers who already have established business models for other solutions must protect their new PaaS business model (see also (Christensen and Overdorf 2000)). 5) Already established software manufacturers should enforce continuous internal use of their PaaS solution.

Table 2. Principles of Form and Function

Design principles	Design options	Design goal	Justificatory knowledge
DP1: PaaS serves at least two distinct customer segments (consuming and developing parties).	Component and application consumers Development partners Individual developers Platform customers	**MR1:** Consider multisidedness	(Drucker 1954), (Johnson et al. 2008), (Osterwalder and Pigneur 2010)

DP2: PaaS core value proposition is to be refined per customer segment and address their "jobs to be done."	Development-focused platform Integration-focused platform Application-based integration platform Distribution channel-based platform	**MR1:** Consider multisidedness **MR2:** Achieve critical mass	(Drucker 1954), (Johnson et al. 2008), (Osterwalder and Pigneur 2010), (Cusumano and Gawer 2002)
DP3: From its launch onwards, the PaaS must offer complementary components and applications.	Develop initial set of components and applications in-house Hire developers Subsidize development partners Actively invest in developers Lower entry and transfer barrier	**MR2:** Achieve critical mass **MR3:** Leverage positive network effects	(Evans 2003), (Eisenmann et al. 2006), (Gawer and Cusumano 2008), (Ceccagnoli et al. 2012)
DP4: Installed base relationships are to be designed to encourage and intensify PaaS usage.	Ensure variety and quality Custom development Integration in roadmap planning Matchmaking capabilities	**MR3:** Leverage positive network effects	(Evans 2003), (Drucker 1954), (Johnson et al. 2008), (Shapiro and Varian 1999)
DP5: Relationships with complementors rely on well-defined rules, IPR regulations, and collaboration models.	Exclusive rights for C&A provider Intellectual property rights and copyrights Integrated into roadmap planning No binding commitments Active search and buyout	**MR3:** Leverage positive network effects **MR4:** Profitability	(Cusumano and Gawer 2002), (Gawer and Cusumano 2007), (Gawer and Cusumano 2008), (Ceccagnoli et al. 2012)
DP6: The internal structure is to be designed in a way that helps avoid conflicts of interest.	Similar goals should be under one executive Organizational culture and processes Internal communication of strategy Shielding of new business model In-house utilization of PaaS	**MR1:** Consider multisidedness	(Cusumano and Gawer 2002), (Christensen and Overdorf 2000)

Expository instantiation

This section illustrates how we applied the design principles in the expository instantiation "for the purpose of theory representation or exposition" (Gregor and Jones 2007). During the two BIE cycles, we designed and evaluated different business model artifacts for Alpha. These artifacts represented Alpha's current business model for SHC, different BMI options, and the updated final version of Alpha's business model for SHC. A representation of the latter is shown in Table 3, and uses the business model framework by Johnson et al. (2008).

Alpha's SHC solution addresses all four identified customer segments: Component and application consumers, development partners, individual developers, and platform customers (DP1). For each customer segment, the "job to be done" has been identified in order to "to solve an important problem or fulfill an important need for the target customer" (Johnson et al. 2008). Alpha's SHC is a development-focused PaaS based on Java (DP2). Currently, Alpha primarily offers integration capabilities with other prominent software solutions provided by Alpha. However, the plan is to further extend integration capabilities towards all internal solutions as well as prominent external solutions of other software providers. Furthermore, Alpha has launched a store for components and applications, in order to also provide a distribution channel for its PaaS customers. One of Alpha's key challenges is to fill the platform with content (DP3), since there was little demand to date for the "empty" platform. Alpha significantly lowered prices for developers as well as its revenue share (now 15%, instead of 30% before). Alpha also offers free trail accounts, has announced developer competitions, and holds events at universities. To date, Alpha has fallen short of leveraging its huge installed base in terms of SHC. To date, Alpha has only established quality assurance and certification processes to ensure high-quality complementary components and applications (DP4). Alpha already started to integrate complementors into its roadmap planning for SHC. However, a final decision on the strategy regarding IPRs and copyrights still needs to be taken and communicated (DP5). Since Alpha is a large software company, establishing an internal organization that supports a multisided business model for SHC is fairly challenging. Alpha has already gone through a re-organization in order to consolidate all teams working on SHC. However, there are still conflicts of interest concerning sales and partner management teams, since these teams prefer to promote solutions that have already proven to be successful. Alpha is currently exploring different ways of shielding the new SHC business model.

Table 3. Business Model of Alpha's SHC

Customer value proposition (CVP)		
Target customer	Job to be done	Offering
Component and application consumers	Use of SaaS applications and/or complement on-premise systems	Application store with certificated applications / Trusted brand / Leverage prior investments through integration capabilities / Hosted platform with a guaranteed availability of 99.9%
Development partners	Develop, deploy, manage, and market business applications and components globally	Large installed customer base / Store for components and applications as distribution channel / Best practices regarding application pricing / Low revenue share rate (15%) / SDK for Java and various integration capabilities / In-memory database technology / Platform guarantees availability of 99.9% / Trusted brand / Support via online community network, FAQ, and documentation
Individual developers	Development of applications with standard tools	Free developer license for noncommercial use / SDK for Java and support of standard Java-based tools / In-memory database technology / Support via online community network, FAQ, documentation, and guides
Platform customers	React cost efficiently to internal and market IT demands; develop applications that can be integrated into legacy systems	SDK for Java and various integration capabilities / In-memory database technology / Platform guarantees availability of 99.9% / Trusted brand / Enterprise readiness regarding SLAs, downtime, etc. / Application and component store / Support via online community network, FAQ, documentation and guides

Key resources	**Key processes**
PaaS platform itself / Infrastructure (data center) for hosting the platform, components, and applications / SDK for Java / Brand / Store for applications and components / Sales partners / Platform technology partners: provide complementing platform features (e.g., mail service) revenue participation	Infrastructure management: 24/7 reliable services, globally available to fulfill the guaranteed availability of 99.9% / Partner management / Efficient operations / Application and component certification process / Provide integration capabilities / Low-cost sales channel (i.e. self-service) / Community network, FAQs, documentation, and guides
Profit formula	
Cost structure	Revenue model
Cost of development, hardware, structured storage, unstructured storage (i.e. documents), backup storage, and resources for applications in staging area / Platform overhead costs / Cost of free instances / Cost for support services / Infrastructure and development costs / Sales and marketing (direct channel)	Platform subscription: EUR 370 to 16,000 per month, or individually composed based on six metrics: virtual machine, unstructured storage, structured storage (IMDB or/and RDMS), bandwidth, and connections to third party systems / Annual fee for development partners: EUR 1,990 / Revenue sharing of 15% for applications/components (offset against annual partner fee) / Application certification fee: initial EUR 990, recurring EUR 495

Discussion and conclusion

This paper sought to present an early stage of our design theory for designing PaaS business models. We employed the DSR paradigm and utilized ADR as research method to build and evaluate our theory. To document our design theory, we adopted the components of a an information system design theory (ISDT) by Gregor and Jones (2007). The purpose and scope of our theory is described by four meta-requirements: 1) Considering multisidedness of PaaS business models. 2) Achieving a critical mass of complementary components and applications as well as consumers. 3) Leveraging positive network effects. 4) Mid-term to long-term profitability of PaaS business models. This paper's main scientific contribution are six design principles that guide PaaS providers in designing effective business models. By proposing a design theory, our research goes beyond previous approaches for studying PaaS: First, the suggested design theory focuses not only on single aspects, such as the revenue model, but addresses the complete set of business model components. Second, it takes a systematic approach for developing prescriptive knowledge about how to design PaaS business models. In doing so, it builds on justificatory knowledge, in particular related to multisided platforms, and develops in continuous building, intervention, and evaluation cycles. Our research is still in progress and thus has certain limitations. Most importantly, the design theory has been developed in close collaboration with one software vendor and has only been instantiated in this organization to date. In terms of Verschuren and Hartog (2005), this means a formative, ex ante and goal-based evaluation of a design plan. Following the evaluation framework by Pries-Heje et al. (2008), we have only performed an ex ante naturalistic evaluation of the design product. However, in order to arrive at a rigorous design theory, further evaluation is needed. Hence, future work will include a naturalistic ex post evaluation and will consider user perspectives.

References

Ballon, P. 2007. "Business modelling revisited: the configuration of control and value," Info (9:5), pp. 6–19.

Bodell, L. 2012. Kill the Company: End the Status Quo, Start an Innovation Revolution, Bibliomotion Inc., p. 256.

Ceccagnoli, M., Forman, C., Huang, P., and Wu, D. J. 2012. "Cocreation of Value in a Platform Ecosystem: The Case of Enterprise Software," MIS Quarterly (36:1), pp. 263–290.

Chesbrough, H. 2007. "Business model innovation: it's not just about technology anymore," Strategy & Leadership (35:6), pp. 12–17.

Christensen, C. M., and Overdorf, M. 2000. "Meeting the Challenge of Disruptive Change," Harvard Business Review (March-April), pp. 1 – 10.

Cusumano, M. 2010. "Technology strategy and management - The evolution of platform thinking," Communications of the ACM (53:1), p. 32.

Cusumano, M. A., and Gawer, A. 2002. "The Elements of Platform Leadership," MIT Sloan Management Review (43:3), pp. 51 – 58.

Drucker, P. F. 1954. The Practice of Management, New York, USA: Harper & Row Publishers.

Economides, N. 1996. "The Economics of networks," International Journal of Industrial Organization (14), pp. 673–699.

Eisenmann, T., Parker, G., and Van Alstyne, M. 2006. "Strategies for Two-Sided Markets," Havard Business Review (October), pp. 1 – 12.

Evans, D. S. 2003. "Some Empirical Aspects of Multi-sided Platform Industries," Review of Network Economics (2:3), pp. 191–209.

Forrester. 2011. "The Forrester WaveTM: Platform- As-A-Service For App Dev And Delivery Professionals , Q2 2011," Cambridge, USA, pp. 1 – 21.

Gartner. 2012. "Hype Cycle for Cloud Computing , 2012," Stamford, Connecticut, USA, pp. 1–84.

Gassmann, O., Frankenberger, K., and Csik, M. 2013. BMI Pattern Cards - Building on 55 business model patterns, St. Gallen, Switzerland: University of St. Gallen, p. 55.

Gawer, A., and Cusumano, M. 2007. "A Strategy Toolkit For Platform Leader Wannabes," in DRUID Summer Conference on Appropriability, Proximity, Routines and Innovation, Copenhagen, Denmark, pp. 1 – 33.

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

Giessmann, A., and Stanoevska, K. 2012. "Business Models of Platform as a Service (PaaS) Providers: Current State and Future Directions," Journal of Information Technology Theory and Application (JITTA) (13:4), pp. 31–55.

Giessmann, A., and Stanoevska-Slabeva, K. 2012. "Platform as a Service – A Conjoint Study on Consumers' Preferences," in Proceedings of the Thirty Third International Conference on Information Systems, Orlando, pp. 1 – 16.

Gregor, S. 2006. "The Nature of Theory in Information Systems," MIS Quarterly (30:3), pp. 611–642.

Gregor, S., and Jones, D. 2007. "The Anatomy of a Design Theory," Journal of the Association for Information Systems (8:5), pp. 312–335.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," MIS Quarterly (28:1), pp. 75–105.

Höfer, C. N., and Karagiannis, G. 2011. "Cloud computing services: taxonomy and comparison," Journal of Internet Services and Applications (2:2), pp. 81–94.

IDC. 2012. "Market Analysis - Worldwide Public Platform as a Service 2012 – 2015 Forecast," Framingham, MA, USA, pp. 1 – 41.

Johnson, M. W., Christensen, C. M., and Kagermann, H. 2008. "Reinventing your business model," Harvard Business Review (December), pp. 51–59.

Kim, W. C., and Mauborgne, R. 2004. "Blue Ocean Strategy," Harvard Business Review (82:10), pp. 76–84.

Mahadevan, B. 2000. "Business models for Internet-based E-commerce : An anatomy," California Management Review (42:4), pp. 55 – 69.

March, S. T., and Smith, G. F. 1995. "Design and natural science research on information technology," Decision Support Systems (15:4), pp. 251–266.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., and Ghalsasi, A. 2011. "Cloud computing — The business

perspective," Decision Support Systems (51:1)Elsevier B.V., pp. 176–189.
Mell, P., and Grance, T. 2011. "The NIST Definition of Cloud Computing," Nist Special Publication, (Vol. 145) Gaithersburg, Maryland: National Institute of Standards and Technology, Information Technology Laboratory, pp. 1–2.
Morris, M., Schindehutte, M., and Allen, J. 2005. "The entrepreneur's business model: toward a unified perspective," Journal of Business Research (58(6)), pp. 726–735.
Osterwalder, A., and Pigneur, Y. 2010. Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, (1st editio.) Amsterdam, Netherlands: OSF.
Osterwalder, A., Pigneur, Y., and Tucci, C. 2005. "Clarifying business models: Origins, present, and future of the concept," Communications of the Association for Information Systems (CAIS) (16:1), pp. 1–25.
Parker, G., and Van Alstyne, M. 2008. "Managing Platform Ecosystems," in Proceedings of the Twenty Ninth International Conference on Information Systems, Paris, pp. 1 – 24.
Pateli, A. G., and Giaglis, G. M. 2004. "A research framework for analysing eBusiness models," European Journal of Information Systems (13:4), pp. 302–314.
Pries-Heje, J., Baskerville, R., and Venable, J. 2008. "Strategies for Design Science Research Evaluation," in 16th European Conference on Information Systems (ECIS), Galway, Ireland, p. 87.
Rochet, J.-C., and Tirole, J. 2003. "Platform Competition in Two-Sided Markets," Journal of the European Economic Association (1:4), pp. 990 – 1029.
Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R. 2011. "Action Design Research," MIS Quarterly (35:1), pp. 37–56.
Shapiro, C., and Varian, H. R. 1999. Information rules: a strategic guide to the network economy, Boston, Massachusetts, USA: Harvard Business Press, p. 352.
Stanoevska-Slabeva, K., and Wozniak, T. 2009. "Cloud Basics - An Introduction to Cloud Computing," in Grid and Cloud Computing: A Business Perspective on Technology and Applications, K. Stanoevska-Slabeva, T. Wozniak, and S. Ristol (eds.), Berlin Heidelberg: Springer, pp. 47–61.
Timmers, P. 1998. "Business Models for Electronic Markets," Electronic Markets (8:2), pp. 3–8.
Tiwana, A., Konsynski, B., and Bush, A. A. 2010. "Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics," Information Systems Research (21:4), pp. 675–687.
Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. 2009. "A Break in the Clouds: Towards a Cloud De nition," ACM SIGCOMM Computer Communication Review (39:1), pp. 50–55.
Verschuren, P., and Hartog, R. 2005. "Evaluation in Design-Oriented Research," Quality & Quantity (39:6), pp. 733–762.
Zhang, Q., Cheng, L., and Boutaba, R. 2010. "Cloud computing: state-of-the-art and research challenges," Journal of Internet Services and Applications (1:1), pp. 7–18.
Zott, C., Amit, R., and Massa, L. 2011. "The Business Model: Recent Developments and Future Research," Journal of Management (37:4)SAGE Publications, pp. 1019–1042.