Abstract

Platform as a Service (PaaS) solutions are changing the way that software is produced, distributed, consumed, and priced. PaaS, also known as cloud platform, offer an execution environment based on software platforms. To be competitive on the market, PaaS providers have to be aware of drivers of successful platforms and design or adjust their business models accordingly. Surprisingly, prior research has made little attempt to investigate consumers’ preferences on PaaS that influence developers’ choice on PaaS solutions. This paper examines this understudied issue through a conjoint study. First a comprehensive literature analysis on PaaS has been conducted in order to build the study design on a rigorous foundation. The conducted conjoint survey contained ten attributes together with 26 corresponding attribute levels and has been completed by 103 participants. Based on the results, a prioritized list of customers’ preferences for PaaS has been created.

Keywords: Platform as a Service, Cloud Platform, Cloud Computing, Business Model


Introduction

Traditionally, software has been developed and distributed by one provider as license-based packaged software. Recently, new phenomena, Platform as a Service (PaaS) and related platform ecosystems have emerged, that are changing the traditional way of software development. Unlike standalone systems and packaged software, PaaS involves transformation of previously highly protected software into platforms and related components or applications (apps) that are developed in emerging ecosystems of third-party developers (Tiwana et al. 2010). PaaS provide a container platform and execution environment wherein third-party developers deploy and run their applications (Rodero-Merino et al. 2011).

This new development paradigm enables platform owners to tap into the benefits of value co-creation and to harness outside expertise and ingenuity on an unprecedented scale (for example there are more than 100'000 developers of iOS (see Tiwana et al. (2010)). It Furthermore, opens up a new revenue stream for platform owners, as they usually require subscriptions from developers and revenue-sharing for sold platform-related modules (see Giessmann et al. 2012 and Eurich et al. 2011). However, providing PaaS requires a shift to at least two-sided business models. Besides selling to end customers, platform owners are encouraging external, third-party developers to develop complementary modules (Ceccagnoli et al. 2012; Tiwiana et al. 2010). Third-party developers are thus becoming a second important customer segment of platform owners.

Given the advantages of platform based development of Internet offerings a trend towards transformation of pure SaaS offerings to platform-based offerings (Giessmann et al. 2012) and from monolith Internet platforms towards PaaS can be observed (see for example Kim et al. (2011)). As a result of these developments the competition of software companies and Internet platforms is increasingly shifting to competition of platform-centric ecosystems and n-sided business models (Giessmann et al. 2012; Tiwana et al. 2010). The increasing competition among platform-centric ecosystems amplifies the battle for external third party developers that get associated to the platform and develop complementary applications.

According to Gartner (2011a), PaaS are currently at the peak of inflated expectations. The market for PaaS is growing fast and will have a potential market volume of more than €9 billion in 2015 (IDC 2012a). Currently, the market is largely fragmented. However, market research companies expect that the distribution of market shares will change dramatically in the next few years (Forrester 2011a; Gartner 2011c; IDC 2012b). It is expected that in the future there will only be a few large providers of cloud platforms, which will offer comprehensive PaaS suites (Gartner 2011b). Providers of PaaS have to further develop their products, as well as their business models, in order to meet the different customers' needs in this fast-growing market.

Despite of their impact and growing importance, PaaS have not yet been considered in a sufficient manner in literature and research (see also Beimborn et al. (2011); Tiwana et al. (2010)). Beimborn et al. (2011) and Tiwana et al. (2010) provide overviews of potential research directions, necessary to build up a deeper understanding of the PaaS phenomena. One important research question proposed by them is the effective platform design from different perspectives. On the one hand the platform design, in particular its features, have impact on the attractiveness of the platform for developers. On the other hand, there are many design options in terms of platform features that platform providers can pursue. Given the highly competitive market for PaaS and the enforced battle for third-party developers, platform owners are facing an important dilemma which PaaS features are most important for third-party developers and need to be prioritized in the design of their platforms. At the same time there is a lack of structured decision support in literature for developers related to the question according to which criteria to choose a platform.

The research presented in this paper contributes to shedding light to the design of PaaS by conducting an empirical investigation on the essential and necessary characteristics of PaaS from the perspective of third-party developers. Thus, the research question can be formulated as follows: what are the preferences of third-party developers, i.e. developing consumers on PaaS? In order to provide a basis from which these research questions can be answered, first a working definition for PaaS is developed based on the existing literature. The literature analysis served also as a basis for identifying an initial list of PaaS features. The initial list of features was discussed, verified and extended in a focus group and prioritized
based on expert interviews. As a result, five PaaS features have been identified as must-have features: availability of 99%; fully automated scalability; standardized Application Programming Interfaces (APIs); high security standards and access control; as well as backup and disaster recovery. To find out what determines the developers’ choice of a PaaS, i.e. to find out the relative importance of PaaS features for developers, an adaptive choice-based conjoint analysis was conducted. The data collection was based on a survey that was completed by more than 100 individuals. The analysis of the data revealed that by far the highest importance was given to the development environment, followed by the test environment and migration among PaaS providers.

The content of the paper is structured as follows: In the next section first a working definition of PaaS is developed and an overview of the state-of-the-art of related research is provided. Then, the applied research methodology is described, followed by a detailed description of the design of the conjoint analysis. In the subsequent section, the empirical results are presented and discussed. Finally, the last section closes the paper with a brief summary and an outlook to further research.

**Definition and State of the Art**

*Towards a Platform as a Service Definition*

Platforms in general can be defined as “a set of subsystems and interfaces that form a common structure from which a stream of related products can be developed and produced efficiently” (Halman et al. 2003). In analogy, platforms in the software industry are referred to 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). While this definition mainly focuses on the extensibility of software platforms in general, the exact definition of PaaS is still open to debate. The following section discusses the Platform as a Service paradigm to achieve a complete definition of what PaaS is, using the main characteristics typically associated with this paradigm in the literature. Consequently, a systematic literature review has been performed, to extract a consensus definition as well as a minimum definition containing the essential characteristics.

In literature PaaS is mainly discussed in context of cloud computing. However, these descriptions focus on just certain aspects of the paradigm. An overview of descriptions of PaaS in literature is provided in Table 1. While the definition of Lawton (2008) reminds of concepts like ASP (application service provisioning), Vaquero et al. (2009), and Rodero-Merino et al. (2011) highlight the execution of third-party components. Mell et al. (2011) as well as Zhang et al. (2010) and Marston et al. (2011) describe PaaS mainly as an abstraction layer from the underlying infrastructure layer including network, servers, operating systems, or storage. In addition to this, Khalidi (2011) and Rimal et al. (2010) mention development, testing, and ongoing maintenance, while Subashini et al. (2011) postulate complete software development lifecycle management.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lawton (2008)</td>
<td>PaaS systems are generally hosted, Web-based application-development platforms, providing end-to-end or, in some cases, partial environments for developing full programs online.</td>
</tr>
<tr>
<td>Vaquero et al. (2009)</td>
<td>... instead of supplying a virtualized infrastructure, they can provide the software platform where systems run on. The sizing of the hardware resources demanded by the execution of the services is made in a transparent manner.</td>
</tr>
<tr>
<td>Rodero-Merino et al. (2011)</td>
<td>PaaS clouds offer an execution environment based on some software platform. [...] A PaaS cloud provides a container platform where users deploy and run their components.</td>
</tr>
</tbody>
</table>
**Mell et al. (2011)**
The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.

**Khalidi (2011)**
PaaS solutions provide a complete application development and hosting site delivered as a cloud service. In addition to managing the underlying infrastructure and offering a metered-by-use cost model, PaaS also facilitates application development, testing, deployment and ongoing maintenance, liberating the customer to focus on managing the application instead of the underlying infrastructure.

**Zhang et al. (2010)**
Built on top of the infrastructure layer, the platform layer consists of operating systems and application frameworks. The purpose of the platform layer is to minimize the burden of deploying applications directly into VM containers.

**Rimal et al. (2010)**
The idea behind PaaS is to provide developers with a platform including all the systems and environments comprising the end-to-end life cycle of developing, testing, deploying and hosting of sophisticated web applications as a service delivered by a cloud based platform.

**Marston et al. (2011)**
A Platform as a Service, or PaaS, facilitates the development and deployment of applications without the cost and complexity of buying and managing the underlying hardware and software layers.

**Subashini et al. (2011)**
PaaS is one layer above IaaS on the stack and abstracts away everything up to OS, middleware, etc. This offers an integrated set of developer environment that a developer can tap to build their applications without having any clue about what is going on underneath the service. It offers developers a service that provides a complete software development lifecycle management, from planning to design to building applications to deployment to testing to maintenance.

Even though the definitions are quite heterogeneous, several common characteristics can be identified: PaaS is a software platform that is opened towards external developers to deploy and run their components. PaaS abstracts from the hardware resources demanded by the execution of the components. Developers no longer have to manage or control the underlying infrastructure, including network, servers, operating systems, or storage – these resources are managed automatically by the platform. Hence, a minimum definition, containing the essential characteristics could be formulated as follows: Platform as a Service refers to an execution environment, wherein external developers deploy and run their complementary components.

In addition to that, the literature analysis revealed some characteristics that seem to be significant for PaaS. Table 2 presents a list of all identified characteristics, along with references that have mentioned the respective characteristic in the context of PaaS.
In order to enable external developers to use PaaS, they are usually provided with a development environment. Development environments can be provided in three different ways. First, a software development kit (SDK) can be provided that allows developers to continue working in their favorite development environment and deploy directly on the platform. Second, a browser-based development environment can be offered. Third, customers are provided with powerful software modeling tools that allow them to create applications on the platform without writing source code. A prominent example of the latter are mashup tools. As a consequence thereof, test and simulation facilities should be provided in order for customers to debug, test, and simulate the developed software components. PaaS solutions facilitate developers to administrate their applications themselves, by providing management tools to start, stop, configure, and/or backup their applications and data. In addition to this, PaaS platforms usually encourage the interactive exchange of knowledge between developers by providing knowledge management facilities. Knowledge management in this context comprises several practices used to identify, create, distribute, and share insights and experiences concerning development in general and the platform in particular.

Based on the conducted literature review and the identified characteristics, the following working definition guides the paper at hand: Platform as a service (PaaS) refers to an execution environment, wherein external developers deploy and run their complementary components. PaaS facilitate the development, testing, and management of software components, as well as knowledge exchange between developers.
State of the Art in Research

The PaaS phenomenon has been considered in literature from different perspectives: As the overview of literature in table 1 and table 2 show, up to approximately year 2010 PaaS was mainly considered as a side topic in literature focusing on the broader phenomenon of cloud computing. Starting from year 2010, literature focusing on PaaS itself started to emerge. Parallel to the emerging scientific literature, market research companies have regularly published market analysis reports regarding cloud technologies.

The early literature on cloud computing that considered PaaS, focused on defining PaaS as part of the overall cloud computing paradigm (for a list of references, see table 1 and 2). Literature focusing on PaaS specifically, considered it from different perspectives: Given that PaaS is quite a new development, a number of articles focused on defining and explaining the features of PaaS as a new paradigm for software development (see for example Buyya et al. (2011); Khalidi (2011); Lawton (2008)). Most of these publications illustrate the PaaS phenomenon with short case studies of prominent examples of existing PaaS. Several authors have furthermore analyzed PaaS as a basis for new n-sided business models and ecosystems in the software industry (see for example Cusumano (2010a); Giessmann et al. (2012); Gonçalves et al. (2011), Iyer et al. (2010)). Other authors have considered specific aspects of PaaS: for example, Eurich et al. (2011) provide an overview of payment models for cloud platforms and Rodero-Merino et al. (2011) analyzed security aspects of PaaS, while Schlagwein (2010) analyzed the degree of openness of existing PaaS. Beimborn et al. (2011) and Tiwana et al. (2010) have identified relevant research questions and future research related to PaaS. Among others, one major topic that was considered as a relevant research question in existing literature is the design of PaaS.

The market research companies focused in their reports on the characterization of the PaaS market in terms of market potential and probable growth (IDC 2010; IDC 2011; IDC 2012a), in terms of structure of the PaaS market (Gartner 2011b; Gartner 2011c) and in terms of market players (Forrester 2011a; Forrester 2011b). In summary, according to market research companies, it can be stated that the PaaS market is a fast-growing market with a potential market volume of more than €9 billion in 2015 (IDC 2012a). The current PaaS market is largely fragmented (Gartner 2011c). However, it is expected that in the future there will be market consolidation towards only a few large PaaS providers that offer comprehensive PaaS suites. Although the market is currently dominated by Microsoft, Salesforce.com and Cordys (Forrester 2011a; Forrester 2011b), it is expected that the distribution of market shares will change dramatically in the coming years, depending on the strategies of current PaaS providers.

In order to sustain themselves in the market, PaaS providers have to adapt their products and especially their business models to the different customer segments, as well as future requirements and opportunities, in this fast-growth market. Thereby, one critical success factor will be to know consumer’s preferences and to address them in PaaS providers’ platform design and business model. While the emerging body of literature on PaaS is starting to shed light on this phenomenon, there is still a lack of knowledge on relevant features of PaaS. From the perspective of platform owners there is a lack of knowledge on how to effectively build PaaS, and from the perspective of third-party developers on how to choose the appropriate PaaS.

Methodology

In order to answer the initial research question - What are developing consumers’ preferences on PaaS solutions? - , a conjoint analysis was carried out. Conjoint analysis (CA) is understood as "[…] a practical set of methods for predicting consumer preferences for multi-attribute options in a wide variety of product and service contexts." (Green et al. 1978, p. 103). The first and possibly most difficult step in designing a conjoint analysis is to identify the features of a product or service - i.e. expressed in the terminology of the conjoint analysis - to identify the attributes that are relevant to consumers in forming their preferences. In line with (Keil et al. 2006) this challenge has been addressed by choosing a three-stage research approach to identify relevant attributes and to design the conjoint survey: (1.) a systematic literature review, (2.) a focus group discussion, and (3.) experts interviews. Figure 1 illustrates our research approach that led to the design of the conjoint study.
An effective literature review served as the basis for preparing an initial list of attributes and attribute levels. A high-quality review “[…] covers relevant literature on the topic and is not confined to one research methodology, one set of journals, or one geographic region” (Webster et al. 2002, p. XV). The keywords “platform as a service”, “PaaS”, “on demand platform”, “cloud platform”, “cloud-based platform” and “cloud computing” have been searched in the following databases: SpringerLink, IEEE, ACM, Springer, Science Direct, Wiley, Informis and AIS Electronic Library (AISeL), and complemented by a Google Scholar search. Thus, according to Levy et al. (2006), most of the important databases, conference proceedings, and journals have been taken into account. Only papers that provided an explicit description of PaaS have been considered within this investigation. Based on these descriptions, PaaS characteristics have been extracted and were mapped to attributes and attribute levels in terms of CA. The initial list contained 19 attributes with a total of 55 attribute levels.

Subsequently, a physical meeting in Spain with eleven, international PaaS experts was organized to discuss the initial list of attributes and attribute levels that were identified based on the literature. Focus groups are a special type of interview involving a small group of interviewees at the same time (Morgan 1998). The advantage of a focus group compared to traditional interviews is the exchange of ideas and points of view among several participants. It allows questions to be expanded and further aspects to be addressed. The objectives of this 1.5 hour long focus group discussion were: (1.) to complete the list of attributes and attribute levels; (2.) to identify must-have/knock-out criteria, meaning attributes that have to be necessarily implemented into PaaS solutions and hence would lead to severe bias within the conjoint analysis (Gustafsson et al. 2007); and (3.) to prioritize the remaining attributes in order to decide which attributes should be included in the conjoint analysis. After the focus group discussion the list was further extended to 21 attributes. However, five attributes have been considered as knock-out criteria.

In order to validate the resulting list of attributes, three qualitative, structured expert interviews were performed (Creswell 2009). The interviewed experts were characterized by having a good overview of the area under investigation and came from both research and industry. They had the following positions/roles: (1.) senior researcher in the software and services area at a large provider of telecommunication and data communication systems; (2.) chief technology officer of a small-medium enterprise in the field of consulting and software development; and (3.) solution platform advisor at a large software company. All three experts had influence on the design and management decisions of PaaS solutions, as well as strategic IT issues, as part of their position/role. Based on the results of the literature analyses and the focus group discussion, the expert interviews had to validate, refine and prioritize the list of attributes. The result of this phase has been the final list conjoint attributes and the corresponding attribute levels, which served as the basis for the actual conjoint study.
The study at hand applies adaptive choice-based conjoint analysis (ACBCA) based on Johnson et al. (2003), which combines the advantages of adaptive- and choice-based procedures. Adaptive conjoint analysis (ACA) asks respondents to perform a self-explicated task where they are asked to rate the individual relevance of every attribute level. If a certain level seems to be completely unacceptable the participant can even exclude it from later questions (Green et al. 1991; Johnson 1987). In contrast to a classical CA or ACA, choice-based conjoint analysis (CBCA) does not ask the respondents to rank product concepts or to rate them on a scale, but rather simulates the process of purchasing a product. For that purpose respondents are shown a set of stimuli/products and asked which one they would most likely buy (Green et al. 2001). Like traditional CBCA, ACBCA asks the participants to choose one product (represented by a stimulus) out of a whole set of alternatives. In addition, ACBCA is able to stabilize estimates using relatively small sample sizes with less than 100 participants, as well as to provide more information from interviews, suitable for part-worths estimations (Johnson et al. 2003; SawtoothSoftware 2009). The relative importance of the attributes and the part-worth utilities are calculated using the Hierarchical Bayes (HB) estimation process (Howell 2009; Orme 2000) using Sawtooth Software\(^1\).

**Conjoint Study**

**Study Design**

**Attributes and Levels**

A list of literature containing about 30 contributions that seem to be relevant to this study resulted from the systemic literature review. Using these articles and papers, the first list of attributes and attribute levels was developed and served as the starting point for the focus group discussion. The initial list contained 19 attributes with a total of 55 attribute levels. After the focus group discussion, the list was further extended to 21 attributes. However, five attributes have been considered as must-have resp. knock-out criteria. The attributes listed in table 3 below have to be necessarily implemented into PaaS solutions in order to succeed in the market in the long run. Due to their importance, these features have to be included in all PaaS solutions. Therefore, they are considered as mandatory and the participants of the ACBCA questionnaire did not have to evaluate them again.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability of 99%</td>
<td>Given that most cloud services are provided across locales and time zones, it is crucial to keep outages and maintenance times to a minimum. An availability of 99% would mean a yearly downtime of 3.5 days in the worst case; anything below that is considered unacceptable.</td>
</tr>
<tr>
<td>Fully Automated Scalability</td>
<td>One of the core features of PaaS in general is the elastic on-demand provision of computing power. It is assumed that it scales automatically to catch up with the demand within the respective service level.</td>
</tr>
<tr>
<td>Standardized APIs</td>
<td>Since most PaaS services do not allow direct access to the operating system, standardized APIs are essential to get access to the platform.</td>
</tr>
<tr>
<td>High Security Standards and Access Controls</td>
<td>Access controls, firewalls, and data encryption are must-haves in order to integrate a PaaS solution into an existing enterprise environment with regulatory compliance and with low security risks.</td>
</tr>
<tr>
<td>Backup and Disaster Recovery</td>
<td>Automated or customizable routines to prevent any kind of data loss.</td>
</tr>
</tbody>
</table>

The remaining attributes were prioritized and the top ten attributes were included in the design of the questionnaire. The final list, which serves as the foundation for the conjoint survey, contains 10 attributes together with 26 corresponding attribute levels and is presented in Table 4 below.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Environment</td>
<td>In order to enable external developers to use the PaaS solution, a development environment can be provided. The development environment can be either online, meaning a completely browser-based development environment offered online, or offline by providing an SDK.</td>
</tr>
<tr>
<td>Test Environment</td>
<td>A test environment facilitates the debugging, testing, and simulation of developed components. A test environment can be either provided or not.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Monitoring of the usage and load of the processes running on the platform can be provided. It can be predefined and just provide a fixed set of views or customizable and let the user specify dynamic notifications, thresholds, etc.</td>
</tr>
<tr>
<td>Mobile Access via App</td>
<td>Cloud platforms can provide a mobile user interface to access the whole development, management, and monitoring functionality; it can be limited to monitoring only or not provided at all.</td>
</tr>
<tr>
<td>Community Features</td>
<td>The PaaS provider can offer a business-related social network for the users and developers of software on the platform, a knowledge-sharing platform, or no community features at all.</td>
</tr>
<tr>
<td>Market Penetration</td>
<td>The market share of a PaaS provider can be high or low.</td>
</tr>
<tr>
<td>Pricing</td>
<td>The pricing model can be revenue sharing, where the PaaS users pay a part of the revenue they make with their software to the PaaS provider, or, more commonly, fixed recurring fees (a flat rate) or a pay-per-use model.</td>
</tr>
<tr>
<td>Marketplace Functionalities</td>
<td>The PaaS provider maintains a marketplace where customers can buy software components. The marketplace can offer provisions for software requests and in addition to that can also offer linkage to other popular marketplaces.</td>
</tr>
<tr>
<td>Payment Handling by</td>
<td>For the marketplace, the PaaS provider could provide the payment infrastructure, or merely usage information.</td>
</tr>
<tr>
<td>Marketplace</td>
<td></td>
</tr>
<tr>
<td>Migration among PaaS</td>
<td>If users decide to migrate their applications to another platform, the PaaS provider can offer migration as a service, provide import/export tools, or none of these.</td>
</tr>
<tr>
<td>Providers</td>
<td></td>
</tr>
</tbody>
</table>

**Questionnaire Design**

The online questionnaire was divided into five sections, see also Figure 1. The first pages of the online survey served as an **INTRODUCTION** by welcoming the participants and providing them with the most important facts regarding platform as a service as well as our definition on PaaS. Furthermore, it requested the demographic data and the professional background of the respondent. Since the goal of the survey was to identify characteristics of PaaS solutions important to consumers, it was necessary that each participant’s employer was already using PaaS or at least planned to invest in such a solution, in order to assure the quality of the choices and statements made during the choice tournament. For this reason, two questions with the aim of sorting out unsuitable participants were inserted.

The first specific ACBCA section contains the so-called **BUILD YOUR OWN** questions. The respondents are asked to indicate their preferred levels of some of the attributes. Based on the answers, the subsequent questions will only contain product concepts including attribute levels that are relatively concentrated around the respondent’s preferred levels (SawtoothSoftware 2009). The build your own questions were
used to reduce the number of error levels during the rest of the survey. Since it did not make sense to include attributes with obvious a priori preferences, only the following attributes have been included in this section: development environment, community features, pricing, payment handling by marketplace, and migration among PaaS providers.

Contrary to traditional CBCA, ACBCA questionnaires do not offer a “none” option during the ‘choice tournament’. The task of the SCREENING section, therefore, is to estimate the “none” parameter threshold by asking the respondents whether they would consider the product concepts shown to be possibilities or not. The screening section of the survey consists of six tasks, each with four product concepts. While the participants answer the questions in the screening section, the software scans their decisions in order to recognize non-compensatory behavior (Johnson et al. 2003). In case the application of such a screening rule can be assumed after the first three screening tasks (meaning that the respondent has systematically avoided/selected an attribute level), the participants are asked the question of whether that level would be completely unacceptable/must have. Attribute levels that have been identified as unacceptable/must have will not be displayed again during the rest of the survey.

The CHOICE TOURNAMENT is the central component of the survey. Based on their answers to previous questions, “[...] respondents are evaluating concepts that are close to their in the build your own section specified product, that they consider ‘possibilities’, and that strictly conform to any cut off (must have/unacceptable) rules” (SawtoothSoftware 2009). As the participants have already indicated which attributes are most important to them by establishing cut-off rules, they can now focus on requirements of secondary importance. A maximum of six sets with three product concepts each is shown to the interviewee. The exact number of stimulus sets shown depends on the concepts marked as possibilities during the screening section.

The last section called CALIBRATION served as an estimate of the part-worth thresholds for “none” by re-showing six concepts, including the concept identified in the build your own section as well as the one winning the choice task tournament and four others. For each concept, the participant is asked how likely he/she is to buy it if it were available in the market, using a five-point scale from "Definitely would not" to "Definitely would" (SawtoothSoftware 2009).

Results

Participants’ background

In order to obtain qualified results the study addressed potential participants, whose employers were already using PaaS or at least planned to invest in using one. Potential suitable candidates were addressed personally by the authors based on contacts resulting from research projects related to cloud computing and PaaS. All potential candidates were asked to share the questionnaire with other suitable persons with similar profile as their own. Due to this “snowball” approach, it remains unclear, how many people exactly have been addressed by the questionnaire. A total of 266 people from Germany (66%), Switzerland (15%), Spain, Italy, Greece, Finland, France, Great Britain, United States and Belgium have completed 103 data sets. The remaining respondents either did not finish the questionnaire (151) or were disqualified because they did not pass the capability assessment (12). The findings presented here are all based on the completed data sets.

Almost 62% of the respondents worked for a company in the software sector, a little more than 4% in the manufacturing industry, more than 11% in service companies, around 2% in governmental institutions, and 21% in other business sectors. The vast majority (82.61%) of the respondents were working for large enterprises with more than 250 employees, 13.04% for small companies with less than 50 employees and 4.35% for medium-sized enterprises with 50-250 employees. Most of the participants considered themselves as employees (77%) and the rest (23%) as members of the management.

2 In cases where the participant does not want to buy any of the presented product concepts/stimuli, he or she can choose the "none" option. This may make the choice situation more realistic but on the other hand bears the risk that indecisive respondents may bypass the decision by choosing "none".
The ages of all the respondents were between 23 and 63, with a mean of approximately 34.4 and a median of 31. Almost 59% of the participants were technically oriented. Not surprisingly, 76% of all the respondents answered the question on how they would rate their technical skills on a scale from 1 (beginner) to 7 (expert) with a value of 5 or higher. However, only 28% ranked themselves with a value of 5 or higher when it came to experience with PaaS. Among those who knew of cloud platform solutions, Google's App Engine was the best known (29.56%), followed by SAP's Business ByDesign (21.73%) and Facebook Developers (12.17%).

Of the evaluated companies, 53.04% already used a PaaS solution and 25.22% planned to invest in PaaS solutions. The remaining 21.74% of the respondents did not know whether their employer was using or planning to invest in PaaS. Of the participants, who knew that the company they were working for was using a PaaS solution or at least planned to invest in one, 31.15% declared that the platform would be used to deploy applications for internal purposes only, 13.11% claimed to work for firms selling or planning to sell the applications, and the remaining 55.74% stated that they develop or plan to develop applications for both internal and external purposes.

**Build Your Own**

The respondents’ answers to the build your own section, where they had to indicate their preferred attribute levels in terms of the development environment, community features, pricing, payment handling by marketplace, and migration among PaaS providers, are displayed in Figure 2. Of the respondents, 57.28% stated that they would prefer an offline development environment, while 38.83% preferred an online one. Less than 4% indicated that they are not interested in a development environment.

The majority of the participants preferred knowledge-sharing tools as a community feature and around 30% stated that they would like to have a business-related social network as part of a cloud platform. A total of 11.65% of the respondents were not interested in community features at all.

With 40.78% of the answers, pay-per-use was the most frequently named preferred pricing model, followed by 39.81% for subscriptions and 19.42% mentioned revenue sharing as their preferred pricing model. More than two-thirds (67.96%) of the participants stated that they would prefer if the marketplace could handle payments and 32.04% would prefer to just get usage information from the marketplace.

A total of 5.83% indicated that they were not interested in migration amongst PaaS providers, while the remaining respondents were almost equally distributed between preferring having migration as a service (47.57) and preferring the PaaS solution to provide tools for migration (46.6%).

![Figure 2. Frequency distribution in ‘build your own’ section](image-url)
Non-compensatory Behavior

Within the screening section, 35.92% of the participants regarded an SDK as a must-have feature of PaaS solutions, while 4.85% regarded an online development environment as a must-have feature. More than 40% indicated that a cloud platform not offering a development environment is unacceptable. Of the participants, 14.56% regarded a test environment as a must-have feature; hence 15 participants regarded not offering a test environment as unacceptable. By contrast, 11.65% of the participants thought that tools for migration among PaaS providers must be offered by a cloud platform and 12.62% of the participants regarded not offering any migration support as unacceptable. In terms of pricing, almost 5% of the participants stated that revenue sharing was not an acceptable pricing model for them.

Relative Importance

The relative importance for each attribute category is depicted in Figure 3. By far the biggest importance (23.84%) was attached to the development environment, followed by the test environment (14.15%) and migration among PaaS providers (12.63%). Pricing (8.79%), mobile device access via app (8.77%) and community features (8.54%) were a little less important to the participants. Market penetration (6.77%), marketplace functionalities (6.32%), payment handling by marketplace (5.38%), and monitoring (4.81%) tailed the field.

![Relative Importance Diagram](image)

Figure 3. Relative importance of attributes based on the part-worth utilities

The importance for each of the features above is an implicit value derived from the absolute range between the highest and the lowest part-worth utility of an attribute. Part-worth utilities for the attribute levels are normalized HB estimates and are depicted in Table 5 and Figure 4.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Level</th>
<th>Part-Worth Utility</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Environment</td>
<td>Online</td>
<td>50.28</td>
<td>73.48</td>
</tr>
<tr>
<td></td>
<td>SDK (offline)</td>
<td>68.73</td>
<td>76.42</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>-119.01</td>
<td>70.21</td>
</tr>
</tbody>
</table>

Table 5. Part-worth utilities of the attribute levels
<table>
<thead>
<tr>
<th>Test Environment</th>
<th>Yes</th>
<th>65.33</th>
<th>50.04</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No</td>
<td>-65.33</td>
<td>50.04</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Customizable (e.g. set thresholds, alarms, etc.)</td>
<td>11.04</td>
<td>24.79</td>
</tr>
<tr>
<td></td>
<td>Predefined</td>
<td>-11.04</td>
<td>24.79</td>
</tr>
<tr>
<td>Mobile Device Access via App</td>
<td>Full Control</td>
<td>27.61</td>
<td>34.86</td>
</tr>
<tr>
<td></td>
<td>Monitoring</td>
<td>8.80</td>
<td>22.10</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>36.41</td>
<td>36.86</td>
</tr>
<tr>
<td>Community Features</td>
<td>Business-Related Social Network</td>
<td>2.60</td>
<td>35.81</td>
</tr>
<tr>
<td></td>
<td>Knowledge Sharing</td>
<td>32.97</td>
<td>36.48</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>-35.57</td>
<td>27.28</td>
</tr>
<tr>
<td>Market Penetration</td>
<td>High</td>
<td>27.26</td>
<td>35.20</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>-27.26</td>
<td>35.20</td>
</tr>
<tr>
<td>Pricing</td>
<td>Revenue Sharing</td>
<td>-21.12</td>
<td>43.13</td>
</tr>
<tr>
<td></td>
<td>Subscription</td>
<td>8.51</td>
<td>49.41</td>
</tr>
<tr>
<td></td>
<td>Pay per Use</td>
<td>12.60</td>
<td>38.29</td>
</tr>
<tr>
<td>Marketplace Functionalities</td>
<td>Marketplace, incl. Provisioning and Linkage to Other Popular Marketplaces (Apple, Android, Blackberry, etc.)</td>
<td>13.00</td>
<td>25.95</td>
</tr>
<tr>
<td></td>
<td>Marketplace, incl. Linkage to Other Popular Marketplaces (Apple, Android, Blackberry, etc.)</td>
<td>11.62</td>
<td>27.51</td>
</tr>
<tr>
<td>Payment Handling by Marketplace</td>
<td>Payment is Handled by Marketplace</td>
<td>11.88</td>
<td>30.28</td>
</tr>
<tr>
<td></td>
<td>Information about Usage Only</td>
<td>-11.88</td>
<td>30.28</td>
</tr>
<tr>
<td>Migration among PaaS Providers</td>
<td>Migration as a Service (by Click or as a Consulting Service)</td>
<td>36.96</td>
<td>35.30</td>
</tr>
<tr>
<td></td>
<td>Tools (e.g. Importing)</td>
<td>28.78</td>
<td>37.51</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>-55.74</td>
<td>34.74</td>
</tr>
<tr>
<td>“None” Option</td>
<td></td>
<td>57.94</td>
<td>139.08</td>
</tr>
</tbody>
</table>

The higher the part-worth utility, the stronger the respondents’ preference was for a certain attribute level. Since normalized part-worth utilities add up to zero for every single attribute, negative part-worth utilities indicate less-desired levels.

In addition to the relative importance of the attributes and the part-worth utilities, hierarchical Bayes estimation generates two statistics indicating the “goodness of fit” for every iteration (Howell 2009). The percentage certainty \(^3\) for the iterations in this data set has a mean of 0.470, indicating a goodness of fit of

\(^3\) This parameter is derived from the likelihood of the data. It indicates how much better the solution is than pure chance. The values of the measure vary between zero (the model fits the data at only the chance level) and one (perfect fit).
approximately 47%. Almost the same result is obtained by looking at the root likelihood (RLH) 4; because each choice task has three alternatives, the RLH for a chance model would be 1/3. The actual value has an average of 0.654; therefore it can be interpreted as two times better than the chance level.

Segmentation

Since the relative importance of the attributes as well as the part-worth utilities of the attribute levels have been calculated on an aggregated level, the values obtained do not predict anything about differences in the behavior of heterogeneous respondent groups. In the following, the participants are therefore divided into different subgroups according to following segmentation criteria: First, by their technical skills, and second, by the size of their company. In order to verify whether the segmentation analyses are valid or not, statistical tests are applied to check the null hypothesis $H_0$ according to which the relative importance of the attributes should not differ among the respondent groups. For both segmentations an independent samples t-test was used. This test "[...] checks whether there are significant differences in the mean values of two groups or not." (Huber et al. 2003).

The first segmentation has been done with respect to the technical skills of the participants. Therefore, the sample has been divided into two groups, one with "very good technical skills" meaning a value of six or seven on a scale from 1 (=Beginner) to 7 (=Expert) (53 participants) and the other with "average or low technical skills" (50 participants). The results show, that in terms of the relative importance of attributes none of the differences between the two groups are significant and therefore $H_0$ cannot be rejected. The only contradictory preferences between technically skilled respondents and average in terms of part-worth utilities appear in the choice of the preferred development environment: Those respondents with

4 This indicator is the geometric mean of the predicted probabilities. A value of one indicates a perfect fit. Like percentage certainty, it is derived from the likelihood.
high technical skills favor SDKs (online: 23.18, offline: 85.7), those with average or low skills the online solutions (online: 75.81, offline: 44.42).

As mentioned in the participants’ background part at the beginning of the chapter, only 9 participants declared that they are working for a small company with less than 50 employees. The data pool for medium size firms is even worse (4 respondents), because the vast majority of the respondents (90 respondents) is working for companies with more than 250 employees. In order to achieve somewhat meaningful segments, participants of SMEs are subsumed in one group. Again, none of the differences between the two groups are significant in terms of relative importance of attribute. However, there are three attribute where the participants disagree on the utility of the corresponding attribute levels: (1.) Community Features: SMEs value knowledge sharing (73.54) and reject Social Network functionalities (-25.28), whereas big companies see both of them positive but with lower importance (25.28 and 6.03). (2.) Pricing: enterprises with more than 250 employees do not want to pay for solutions by sharing their revenue (-25.80), while SMEs do have positive utility (4.57). But participants of small or mid-size firms prefer a pay per use model (20.23) whereas their counterparts prefer a monthly flat-fee (14.01). (3.) From the perspective of big companies, the marketplace only has to be linked to other popular stores (12.77) and not necessarily also have to provide provisioning (8.87) whereas SMEs clearly favor "all-inclusive" marketplaces (28.05).

Limitations

Despite of the significant results, the study has several limitations that might affect the generalizability of the results and that need to be mentioned. First of all the study was based on a small sample of 103 completed questionnaires. This number is sufficient to assure results with statistical significance for the sample, but limits the generalizability of the study. Furthermore, most of the participants are based in Germany and Switzerland, thus the study mainly reflects the preferences of German speaking PaaS developers. Customers from regions, where the usage of PaaS is higher and more mature than in Europe (e.g. according to IDC (2012a), 64.5% of the PaaS market is in the USA), might have slightly different preferences. Moreover, the vast majority (82.61%) of the respondents were working for enterprises with more than 250 employees. This limits the study as well, since on the hand SME’s are clearly underrepresented, and on the other hand it remains unclear how large those enterprises really are. Thus, additional research is required to verify and triangulate the achieved results, by considering samples of participants representing different regions, company sizes, software development approaches, cultures and different customer segments (i.e. existing and potential PaaS customers, or customers focusing on external or internal deployment of applications).

Interpretation

Based on the outcomes of the literature review, the focus group discussion, the expert interviews, and the ACBCA survey, today's favored PaaS solutions should include the following features: availability of at least 95%, fully automated scalability, standardized APIs, high security standards and access control, and backup and disaster recovery. These are clearly the must-have features of PaaS solutions.

A development environment that is provided either online (50.28) or preferably as an SDK (68.73) can also be considered as a must, especially since 40% of the participants indicated that a cloud platform not offering a development environment is unacceptable. With a relative importance of 14%, the same is true for the availability of a test environment, which was also indicated to be a must-have feature by 15 participants. With a relative importance of 13%, smooth migration among PaaS providers is of high significance to consumers. Thereby, consumers prefer PaaS solutions to offer the migration itself as a service (36.96), not just provide tools to support the migration process (28.78). In addition to this, 13 participants explicitly stated that not providing migration support is non-compensatory for them.

Although in the literature it is often claimed that PaaS is characterized especially by a pay-per-use model (see Armbrust et al. 2010; Khalidi 2011; Lindner et al. 2010; Vaquero et al. 2009; Weinhardt et al. 2009), this cannot be confirmed from a consumer perspective. Although pay-per-use models are somewhat popular (8.51), most of the respondents preferred to pay for their solution on a monthly flat-fee basis (12.60). Revenue sharing was considered least attractive (-21.12). Also, 19.42% of the respondents
mentioned revenue sharing as their preferred pricing model within the build your own section of the questionnaire. These results seem somehow contradictory. One could assume that there are other factors influencing the preferred pricing model, like for instance the type and focus of the PaaS platform itself.

Contrary to the assessments of the experts interviewed, mobile device access obtained a relative importance of more than 9%. Although all the experts agreed that mobile device access to a cloud platform would be less significant, consumers clearly prefer to have full control (27.61) via a mobile device instead of having only monitoring features available on mobile devices (8.80). With relative importance of 9%, consumers do have a distinct interest in community features. However, they clearly focus on knowledge-sharing features (32.97) and are less interested in socially enhanced features (2.60).

**Discussion**

The goal of the research presented in this paper was the assessment and prioritization of consumers’ preferences on PaaS. The research provided the following results: First, the available emerging literature on PaaS was summarized and the term PaaS was defined. Then, grounded on a combination of literature research, focus groups and expert interviews, a list of potential features of PaaS was developed. The identified features were classified in must-haves (see table 3) and other relevant features (see table 4). The relevance and priority of the other features for customers of PaaS was assessed with an online survey and analyzed based on adaptive choice-based conjoint analysis. The analysis revealed the relative importance of the analyzed features for PaaS platforms (see figure 3 and table 5).

According to the analysis, the must-have features are: availability of 99%; fully automated scalability; standardized Application Programming Interfaces (APIs); high security standards and access control; as well as backup and disaster recovery. These features can be considered as knock-out criteria from the perspective of external developers. Platform owners need to provide the must-have features in order to enable a reliable and trustable development environment for external developers and to decrease their risk in using an external development and deployment environment. Furthermore, in particular, a fully automated scalability reflects particularly the specific and innovative advantage of using PaaS, while standardized APIs provide the basis for increased effectiveness of external developers.

From the remaining ten attributes considered in the analysis, the highest relevance was attributed to the development environment, the testing environment and the migration among PaaS. These findings taken together with the must-have PaaS features enforce the developer point of view on PaaS. The least priority was attributed to market and business related aspects of Paas offerings, as market penetration, marketplace functionality and payment handling by the PaaS marketplace. These findings are to some extent surprising, as it can be expected that developers are interested not only to develop complementary applications, but also profit from the platform support for selling their applications. However, these findings reflect that the majority of the respondents are technically oriented and most probably responsible only for development and maybe not for marketing and distribution of the components. Middle or moderate importance was attributed to pricing, mobile device access via app and community features. The relatively high importance of mobile access and of community features adds to the developer point of view on PaaS. A PaaS should be accessible anytime, and from anywhere. Furthermore, Paas are considered to be important knowledge exchange environments for developers.

The segmentation analysis revealed that differences in setting priorities might arise depending on the size of the company involved or the differences of skills of involved developers. Thus, PaaS owners need to know who their customers are and diversify their offerings according to the specific needs of identified customer segments.

With these results, the paper at hand provides a significant scientific and practical contribution. The scientific contribution consists of the following: The paper provides first an overview and summary of emerging PaaS literature and available PaaS descriptions. This contributes to a clear definition of PaaS and its delimitation of other related concepts as cloud computing. Furthermore, the identified must-have features and the prioritized list of additional features provide a significant contribution to a deeper understanding of the widely unexplored phenomenon PaaS (see also Beimborn et al. (2011); Tiwana et al. (2010)). These results provide also a contribution to a design theory for PaaS by considering the customer perspective. Finally, from a scientific point of view the paper demonstrated the applicability of the
methodology mix consisting of literature research, focus groups and expert interviews to identify and triangulate an initial list of potential attributes for phenomena under investigation and their subsequent prioritization based on an online survey and conjoint analysis. The same methodological approach can be applied for similar research problems as part of design science approaches, where the priority of features for software artifacts is not given, but requires input from customers.

From a practical point of view the results presented in the paper at hand provided a valuable contribution to providers of PaaS and to customers of PaaS. PaaS providers are active in a market with high growth potential but also with competition and high probability of consolidation in the future. In order to sustain in such a market, PaaS providers need to quickly improve their PaaS solutions. The priority list of required attributes provides valuable input in which features to invest first. The resulting preference list suggests that PaaS owners need to provide a secure, reliable and scalable development and testing environment, which is also accessible over the mobile channel and provides community features. For PaaS customers the resulting list of must-have and prioritized features can serve as a checklist for assessing and evaluating of available PaaS solutions.

Conclusion & Outlook

The purpose of the current study was to determine consumers’ preferences on PaaS solutions. In that regard, platform as a service has been defined as execution environments wherein external developers deploy and run their components. PaaS also facilitate the development, testing, and management of software components, as well as the exchange of knowledge between developers. It was shown that the PaaS market is a fast-growing market with a huge potential market volume, but still very fragmented at present. This study has shown that PaaS solutions that aim to attract software developers have to fulfill certain market needs and expectations. In particular, PaaS solutions should contain a sophisticated development environment (online or offline), offer a service for the importing of data and/or software during migration among different PaaS providers, and include a reliable test environment. They furthermore need to meet prevailing security, reliability, scalability, standardization and mobile access requirements. From the perspective of potential developers, the resulting list of prioritized features can serve as an evaluation checklist for PaaS.

The research presented in the paper at hand contributes to the existing PaaS knowledge. The identified limitations of the analysis and remaining open questions provide starting points to further research. The relatively low importance of business and market place oriented requirements can be verified in additional studies that explicitly involve also persons involved in marketing and sales of developed complementary platform-based applications. Furthermore, the results of the segmentation analysis suggest that there might be differences in the preferences of large and small and medium sized companies as well as depending on the level of technical skills of persons involved in the analysis. Thus, additional studies are necessary where each customer segment might be considered explicitly.

Acknowledgements

The research leading to these results has partially received funding from the 4CaaSt project (http://www.4caast.eu/) from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 258862. This paper expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this paper.

References


