What are Developers’ Preferences on Platform as a Service? An Empirical Investigation

Andrea Giessmann
SAP Research Center St. Gallen
andrea.giessmann@sap.com

Katarina Stanoevska-Slabeva
University of St.Gallen
katarina.stanoevska@unisg.ch

Abstract

Platform as a service (PaaS) solutions are changing the way how software is produced, distributed, consumed, and priced. Due to the wide variety of approaches to PaaS and the absence of a formula that represents the drivers of successful platforms, the goal of this paper is to identify developers’ preferences on PaaS. An adaptive choice-based conjoint analysis has been performed and was completed by 103 participants. Based on the results, a prioritized list of developers’ preferences for PaaS has been created. The results of the survey indicate that, third-party developers’ preferences also depend on the phase of the software lifecycle in which they are (Development, Distribution, and Operation).

1. Introduction

Platform as a service (PaaS) solutions are changing the way how software is produced, distributed, consumed, and priced. PaaS provides a container platform and execution environment wherein third-party developers deploy and run their applications [1]. This new development paradigm enables platform owners to tap into the benefits of value co-creation and to leverage outside expertise [2]. Due to this advantage a tendency towards transformation of both pure SaaS and IaaS offerings and monolith Internet platforms towards PaaS can be observed, see [3]. As a result, the competition of software companies and Internet platforms is increasingly shifting to competition of platform-centric ecosystems [2], [3]. Providers of PaaS are under pressure to further develop their products, in order to sustain on the market and meet the different customers’ needs.

Despite of their impact and growing importance, PaaS have not yet been considered in a sufficient manner in literature and research (see also [2], [4]). [2] and [4] suggest 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. The features of the platform have major impact on its attractiveness for external developers and their interest to contribute to the platform. Thus, the need for knowledge related to the question how to build and establish PaaS effectively is growing.

The research presented in this paper contributes to shedding light to the design of PaaS by conducting an empirical investigation on their essential requirements 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. consumer on PaaS? To answer this question, first a working definition for PaaS is provided based on the existing literature. To find out what determines the choice of a PaaS by developers; a conjoint analysis as well as a factor analysis have been conducted, which are described in chapter 4. After presenting the results of this study, they will be discussed afterwards in chapter five. Finally, the last section closes the paper with a brief summary and an outlook to further research.

2. Definition and state of the art

2.1. Definition of platform as a service

[2] defines a platform as a “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”. While the general definition provides a broad selection of what a software platform might be, the exact definition of PaaS is still open to debate.

PaaS is discussed in literature mainly in context of cloud computing [1], [2], [5–11]. 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 [1], [5–17]. PaaS
abstracts from the hardware resources demanded for 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 PaaS [6–8], [11], [15]–[18].

In order to enable external developers to use PaaS, a development environment that is accompanied with test and simulation facilities is provided to enable developers to debug, test, and simulate the developed software components [5], [10–12], [15], [17], [19]. PaaS solutions facilitate developers to administrate their applications themselves, by providing management tools to start, stop, configure, and/or backup their applications and data [5], [10], [12], [15], [17]. 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 [5], [19]. Paas are furthermore increasingly enhanced with marketplaces for applications of third-party developers [3].

Based on the conducted literature review the following working definition guides the paper at hand: PaaS refers to an execution environment, wherein external developers deploy and run their complementary components. PaaS facilitate the development, testing, deployment, execution, management and distribution of software components, as well as knowledge exchange between developers.

2.2. State of the art in PaaS research

The PaaS phenomenon has been considered in literature from different perspectives: The early literature focused on defining PaaS as part of the overall cloud computing paradigm (see i.e. [5–17]). 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 [5], [8], [20]). 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 [3], [21], [22]). Other authors have considered specific characteristics of PaaS: i.e. [23] provide an overview of payment models, [1] analyzed security aspects, and [24] analyzed the degree of openness of existing PaaS. [2], [25] have identified relevant research questions and future research related to PaaS. One major topic that was considered as a relevant research question is the design of PaaS.

The market research companies focused in their report on the characterization of the PaaS market in terms of: market potential and probable growth [26], structure of the PaaS market [27], and market players [28]. Although the market is currently dominated by Microsoft, Salesforce.com and Cordys, it is expected that the distribution of market shares will change dramatically in the coming years, depending on the strategies of current PaaS and emerging providers [28]. 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-growing market. Thereby, one critical success factor will be the specific design of the platform [2], [4].

3. Methodology

In order to answer the initial research question - What are third-party developers’ preferences on PaaS? a conjoint analysis (CA) is carried out. Conjoint analysis 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" [29]. The paper at hand applies the variant of an adaptive choice-based conjoint analysis (ACBCA) [30], which combines the advantages of adaptive- and choice-based procedures. Results based on the same dataset have previously been published in [31].

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 [32], [33]. 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 [34]. Like traditional choice-based conjoint analysis (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 [30], [35]. The relative importance of the attributes and the part-worth utilities are calculated using the Hierarchical Bayes (HB) estimation process [36], [37] using Sawtooth Software.

The first step in designing a conjoint study is to identify the attributes that are relevant to consumers in forming their preferences. To identify requirements of PaaS consumers, that can be mapped to attributes and attribute levels in terms of conjoint analysis later on, a three-stage research approach was chosen. First, a systematic literature review, which followed the methodology proposed by [38], served as the basis for preparing an initial list of requirements. The identified requirements of PaaS are later mapped to attributes and attribute levels in terms of conjoint analysis. The keywords “platform as a service”, “PaaS”, “on demand platform”, “cloud platform”, “cloud-based platform” and “cloud computing” have been search in the following databases: SpringerLink, IEEE, ACM, Springer, Science Direct, Wiley, Informs and AIS Electronic Library (AISeL), and complemented by a Google Scholar search. Only papers that provided an explicit description of PaaS have been considered within this investigation.

Second, a focus group discussion was organized. The initial list of identified requirements based on literature were discussed by 11 PaaS experts with the objectives to: (1.) complete the list of requirements, resp. attributes and attribute levels; (2.) identify knock-out criteria, meaning requirements that have to be necessarily implemented into PaaS solutions and hence would lead to severe bias within the conjoint analysis; and (3.) prioritize the remaining requirements in order to decide which requirements should be included in the conjoint analysis.

In order to validate the resulting list of requirements, three qualitative, semi-structured expert interviews were performed. 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, the focus group discussion, and the expert interviews, the questionnaire for the empirical investigation was designed, by mapping the identified requirements to attributes and attribute levels.

In addition to the conjoint analyses the variability among the attributes has been investigated by means of an exploratory analysis. A factor analysis was used in order to extract a small number of relevant mutually independent factors from the original number of attributes [39]. Two discrete classes of factor analysis, namely exploratory factor analysis (EFA) and confirmatory factor analysis (CFA), can be distinguished [40]. While in CFA it is required to “…have a firm a priori sense, based on past evidence and theory, of the number of factors that exist in the data, of which indicators are related to which factors…” [41] within an EFA the researcher may not have any specific expectations regarding the number of underlying factors. As the research at hand is of exploratory nature, EFA was applied.

4. Empirical Investigation

4.1. Study Design

4.1.1. Attributes and level. The systemic literature review resulted in 30 contributions that seem to be relevant to this study. Based on them, the first list of requirements was developed and served as the starting point for the focus group discussion. The initial list for the conjoint analysis contained 19 attributes with a total of 55 attribute levels. After the focus group discussion and the expert interviews, the list was further extended to 21 attributes. However, 5 attributes have been considered as knock-out criteria by the interviewed experts: (1.) Availability of at least 99%; (2.) Fully automated scalability; (3.) Standardized APIs; (4.) High Security Standards and Access Control; and (5.) Backup and Disaster Recovery. These attributes are “must-haves” and have to be necessarily implemented into PaaS solutions in order to assure success in the market in the long run. Therefore, they are considered as mandatory and the participants of the ACBCA questionnaire did not have to evaluate them again.

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 ten attributes together with 26 corresponding attribute levels and is presented in Table 1 below.
questions will only contain product concepts including attribute levels that are relatively concentrated around the respondent's preferred levels [35]. 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 [30]. 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" [35]. 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" [35].

### Table 1. Conjoint attributes

<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 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 Functionali- ties</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 Marketplace</td>
<td>For the marketplace, the PaaS provider could provide the payment infrastructure, or merely usage information.</td>
</tr>
<tr>
<td>Migration among PaaS Providers</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>
</tbody>
</table>

### 2.1.2. Questionnaire design

The questionnaire was designed to reflect the various preference evaluation steps of an ACBCA analysis, plus an introductory section: The introductory section was dedicated to selecting survey participants with required characteristics and collecting their demographic data. Given the special focus of the analysis, 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
4.2. Results

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 addresses 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.

4.2.1. Participants background. 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.

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 such an on-demand solution. 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.

4.2.2. Results of the Conjoint Analysis. 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 as follows: 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 PaaS. 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% of responses for subscriptions and 19.42% for the revenue sharing 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%).

Within the Screening section, 35.92% of the participants regarded an SDK as a must-have requirement of PaaS solutions, while 4.85% regarded an online development environment as a must-have requirement. More than 40% indicated that PaaS not offering a development environment is unacceptable. Of the participants, 14.56% regarded a test environment as a must-have requirement; 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 1. 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.

![Figure 1. Relative importance of attributes](image)

The importance for each of the requirements 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 2 and Figure 2. 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 [36]. The percentage certainty for the iterations in this data set has a mean of 0.470, indicating a goodness of fit of approximately 47%. Almost the same result is obtained by looking at the root likelihood (RLH): 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.

### Table 2. Part-worth utilities

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></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>
<tr>
<td>Test Environment</td>
<td>Yes</td>
<td>65.33</td>
<td>50.04</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>-65.33</td>
<td>50.04</td>
</tr>
</tbody>
</table>

### 4.2.3. Interpretation of the Conjoint Analysis.

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 address the following requirements: 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 requirements of PaaS solutions.

A development environment that is provided either online (50.28) or preferably as a SDK (68.73) can also be considered as a must, especially since 40% of the participants indicated that a PaaS 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 requirement by 15 participants. With a relative importance of 13%, smooth migration among PaaS providers is of high significance to consumers. Therefore, consumers...
prefer PaaS solutions to offer the migration themselves as a service (36.96), not just provide tools to support the migration process (28.78). In addition, 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 [6], [8], [12], 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 example 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 PaaS 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).

4.2.4. Results of the Exploratory Factor Analysis.

Principal components analysis (PCA) is used as factor extraction method in the paper at hand. However, in order to fulfill the normal distribution assumption on which the correlation matrix is based on, the following attributes have been transformed by squaring: Test environment, marketplace functionalities, monitoring and market penetration. In order to determine the factors, two statistical verification methods are applied, the Kaiser-Guttman criterion and the scree plot test. According to the Kaiser-Guttman criterion the number of factors to be extracted should equal the number of factors with eigenvalues larger than 1 (cf. Table 3). This results in the extraction of five factors explaining 73.34% of the total variance. The scree plot points to the same result.

As final step the nature of the underlying constructs has been clarified by applying the Varimax method as the most common factor rotation method with Kaiser normalization [40]. The rotated component matrix is depicted in Table 4. Items are assigned to a factor if the factor loading adds up to at
least 0.55 [39]. The five factors vary in the number of items with a range from three to one attribute items.

Table 3. Eigenvalues

<table>
<thead>
<tr>
<th>Factor</th>
<th>Total</th>
<th>% of Variance</th>
<th>Cumulative %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.241</td>
<td>22.411</td>
<td>22.411</td>
</tr>
<tr>
<td>2</td>
<td>1.480</td>
<td>14.797</td>
<td>37.208</td>
</tr>
<tr>
<td>3</td>
<td>1.389</td>
<td>13.894</td>
<td>51.102</td>
</tr>
<tr>
<td>4</td>
<td>1.209</td>
<td>12.088</td>
<td>63.189</td>
</tr>
<tr>
<td>5</td>
<td>1.015</td>
<td>10.151</td>
<td>73.340</td>
</tr>
<tr>
<td>6</td>
<td>.839</td>
<td>8.394</td>
<td>81.734</td>
</tr>
<tr>
<td>7</td>
<td>.691</td>
<td>6.908</td>
<td>88.642</td>
</tr>
<tr>
<td>8</td>
<td>.594</td>
<td>5.944</td>
<td>94.586</td>
</tr>
<tr>
<td>9</td>
<td>.515</td>
<td>5.148</td>
<td>99.734</td>
</tr>
<tr>
<td>10</td>
<td>.027</td>
<td>.026</td>
<td>100.000</td>
</tr>
</tbody>
</table>

Table 4. Results of factor analysis

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Factor 1</th>
<th>Factor 2</th>
<th>Factor 3</th>
<th>Factor 4</th>
<th>Factor 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration among PaaS Providers</td>
<td>0.849</td>
<td>.005</td>
<td>.137</td>
<td>-0.121</td>
<td>.084</td>
</tr>
<tr>
<td>Development Environment</td>
<td>-0.034</td>
<td>-0.017</td>
<td>0.009</td>
<td>-0.013</td>
<td>-0.012</td>
</tr>
<tr>
<td>Mobile Device Access via App</td>
<td>-0.582</td>
<td>-0.214</td>
<td>-0.013</td>
<td>0.522</td>
<td>-0.099</td>
</tr>
<tr>
<td>Test Environment</td>
<td>-0.064</td>
<td>-0.037</td>
<td>.069</td>
<td>-0.089</td>
<td>.159</td>
</tr>
<tr>
<td>Payment Handling by Marketplace</td>
<td>-1.011</td>
<td>-0.065</td>
<td>-0.107</td>
<td>.035</td>
<td>-0.401</td>
</tr>
<tr>
<td>Marketplace Functionalities</td>
<td>0.215</td>
<td>0.652</td>
<td>-0.201</td>
<td>.233</td>
<td>-0.131</td>
</tr>
<tr>
<td>Monitoring</td>
<td>-0.043</td>
<td>-0.081</td>
<td>-0.088</td>
<td>.294</td>
<td>.248</td>
</tr>
<tr>
<td>Community Features</td>
<td>-0.193</td>
<td>0.105</td>
<td>0.646</td>
<td>0.246</td>
<td>0.028</td>
</tr>
<tr>
<td>Market Penetration</td>
<td>-0.005</td>
<td>-0.035</td>
<td>-0.066</td>
<td>-0.898</td>
<td>-0.127</td>
</tr>
<tr>
<td>Pricing</td>
<td>0.012</td>
<td>0.071</td>
<td>-0.041</td>
<td>-0.105</td>
<td>0.067</td>
</tr>
</tbody>
</table>

Rotation in 7 iterations converged.

4.2.4. Interpretation of the Factor Analysis. The five factors consist of three to one attribute. Factors consisting of only one attribute (Factor 3 = Market Penetration and Factor 4 = Pricing) are not considered further, since these attributes have already been introduced and discussed previously.

Factor 1: Development phase. The first factor is made up of three attributes which relate to the development phase in the software development lifecycle. While migration among PaaS providers and mobile device access via app load positively on the factor, development environment loads negatively. The positive and negative loads can be explained by the initial situation of the respondents. For instance, respondents that have already developed a software component have a strong interest in migration support, while respondents planning to develop software components require development environments.

Factor 2: Distribution phase: The second factor is made up by three attributes which related to the distribution phase of the developed components. Marketplace functionalities as well as payment handling provided by the marketplace have positive loads on the distribution factor, while not providing a test environment has a clear negative influence on the distribution phase. PaaS consumers need to be able to test and demonstrate their software components in order to sell them.

Factor 3: Operation phase: Two attributes exhibit high positive loadings on the third factor: Monitoring and community features. In summary, they can be characterized by the term “Operations”, since both requirements become important once the developers have already decided on the platform and are actively working with the PaaS solution. In this phase of the software component lifecycle developers need to exchange knowledge and experience related to the platform as well as monitor their developed software components.

5. Discussion

The goal of the research presented in this paper was the assessment and prioritization of developers’ preferences on PaaS. The research provided the following results: Grounded on a combination of literature research, focus groups and expert interviews, a list of potential PaaS requirements (latter mapped to attributes and attribute levels in terms of CA) was developed. The identified requirements were classified in must-haves and other relevant requirements (see Table 1). The relevance and priority of the other requirements for PaaS developers was assessed with an online survey based on ACBCA. The ACBCA analysis revealed the relative importance of the analyzed PaaS requirements (see Figure 1 and Table 2). The attributes were furthermore analyzed with a factor analysis. The factor analysis indicated, three main phases that might influence developers’ preferences during the software lifecycle: Development, Distribution or Operation. The results of the factor analysis provide an indication that PaaS developers might prioritize different requirements depending on their phase in the software lifecycle.

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. Furthermore, the identified must-have requirements and the prioritized list of additional requirements provide a significant contribution to a deeper understanding of the widely unexplored phenomenon PaaS (see also [2], [25]). These results provide also a contribution to a design theory for PaaS that is based on consideration of the customer perspective. Finally, 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 requirements. The same methodological approach can be applied for similar research problems as part of design science research.
approaches, where the priority of requirements for software artifacts is not known, but requires input from consumers.

From a practical point of view the results presented in the paper at hand provided a valuable contribution to customers and providers of PaaS. For PaaS customers the resulting list of must-have and prioritized requirements can serve as a checklist for assessing and evaluation of available PaaS solutions. PaaS providers are active in a market with high growth potential but also with high competition. In order to sustain in such a market, PaaS providers need to quickly improve their solutions. The priority list of required attributes provides valuable input in which requirements to invest first. Furthermore, the three factors resulting from the factor analysis indicate that preferences of developers might vary depending on the phase of the software lifecycle, as well as the application domain. For example, customers focusing on deploying components for internal purposes might stress more attributes that define the development factor, while those that consider also external distribution might have higher preferences for attributes defining the distribution factor. However, this is not clearly reflected in the results of the study. Thus, different customer segments might be considered: customers that use the PaaS already and new customers or customers using Paas for internal or external distribution.

6. Conclusion & Outlook

The purpose of the current study was to determine developers’ preferences on PaaS solutions. PaaS has been defined as an execution environment, wherein external developers deploy and run their complementary components. PaaS facilitate the development, testing, deployment, execution, management and distribution of software components, as well as the exchange of knowledge between developers. At present the PaaS market is considered to be a fast-growing market with a huge potential market volume. Hence, the current market for PaaS offers great opportunities for the introduction of innovative solutions. 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 tools that support migration among different PaaS providers, as well as include a reliable test environment.

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, were the usage of PaaS is higher and more mature than in Europe (e.g. according to [26], 73% 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).

7. References