8-5-2011

USER-DESIGNED SOFTWARE AS A SERVICE – CAN CONCEPTUAL WORK BE SHIFTED TO THE END-USER

Jochen Hetzenecker
Friedrich Alexander University Erlangen-Nuremberg, jochen.hetzenecker@wiso.uni-erlangen.de

Martin Weiner
Friedrich-Alexander Universitat Erlangen-Nurnberg, martin.wiener@wiso.uni-erlangen.de

Michael Amberg
Friedrich Alexander University Erlangen-Nuremberg, michael.amberg@wiso.uni-erlangen.de

USER-DESIGNED SOFTWARE AS A SERVICE – CAN CONCEPTUAL WORK BE SHIFTED TO THE END-USER

Jochen Hetzenecker
Friedrich Alexander University Erlangen-Nuremberg
jochen.hetzenecker@wiso.uni-erlangen.de

Martin Wiener
Friedrich Alexander University Erlangen-Nuremberg
martin.wiener@wiso.uni-erlangen.de

Michael Amberg
Friedrich Alexander University Erlangen-Nuremberg
michael.amberg@wiso.uni-erlangen.de

ABSTRACT

Cloud Computing and especially Software as a Service (SaaS) is predicted to bring about the next information technology (IT) revolution. Benefits and risks have been discussed extensively in both research and practice. Key benefits like higher flexibility and lower costs are confronted with major risks such as loss of data control. These benefits and risks are predominantly discussed in the context of traditional SaaS. Here, the administrator configures and maintains the application. In contrast, new SaaS concepts allow shifting conceptual work to end-users by enabling them to adjust standard SaaS software independently from their IT department. Using an action research approach, we investigated the benefits and risks of this so called user-designed SaaS, and compared them to those of traditional, administrator-designed SaaS. Many benefits and risks still apply to user-designed SaaS, but also so far unknown benefits like higher user acceptance and lower dependency to administrators, as well as new risks like shifting conceptual skills to the end-user were observed.

Keywords

Cloud Computing, Software as a Service, Google Apps, Action research, Software development.

INTRODUCTION

Since 2007, more and more attention has been given to Cloud Computing (Armbrust, Fox, Griffith, Joseph, Katz, Konwinski, Lee, Patterson, Rabkin, Stoica and Zaharia, 2010). "Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services)" (Vaquero, Rodero-Merino, Caceres and Lindner, 2009). In 2009 and 2010, Gartner's Hype Cycle placed Cloud Computing at the peak of expectations (Gartner Inc., 2010). Software as a Service (SaaS) as the most used part of Cloud Computing depicts the usage of software on-demand as a service (BITKOM, 2009). It was first mentioned in 2001 (Software & Information Industry Association, 2001), but had its breakthrough first in the Cloud Computing context. Studies predict that SaaS revenues will grow six times faster than other software services (Wainewright, 2009), and that the SaaS turnover rate will increase tenfold in Germany from 2010 to 2015 (BITKOM, 2009).

During the last years, the benefits and risks have been discussed in research and practice in the context of "traditional" SaaS solutions still requiring an administrator to install, configure and operate the application. More recently, new SaaS concepts were introduced to the market. In contrast to traditional concepts, some new SaaS concepts allow to shift conceptual work to end-users by enabling them to adjust standard SaaS software independently from their IT department. The end-user has the possibility to create and modify software by himself, without the help of an administrator. Therefore, a shift of the conceptual tasks from the administrator to the end-user is required, with the latter having to develop new skills. Through adoption and combination of tools like calendars, emails, websites, and documents, software tools like guidance systems, tracking lists, or information platforms can be created.

By now, we are unaware of any study examining the benefits and risks of this new type of SaaS. To address this gap in prior research, we formulated the following research questions: Do benefits and risks of traditional, administrator-designed SaaS

scale to new, user-designed SaaS? Which unknown benefits and risks occur? To examine these research questions, we conduct action research in a German university.

THEORETICAL BACKGROUND

Cloud Computing and Software as a Service

In this paper, Cloud Computing is defined as "a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services)" (Vaquero et al., 2009). According to Lawton (2008), "users work with web-based, rather than local, storage and software" which offers certain advantages. Hence, the increase of flexibility is one of the most important opportunities of Cloud Computing. You can get access to 100,000+ processors only by having a credit card (Weinhardt et al., 2009).

Cloud Computing encompasses a wide range from virtualized hardware to software running in a website. Due to a better separation and categorization, three subparts are defined (Weinhardt et al., 2009): Infrastructure as a Service (IaaS) hereby describes the hardware view. It addresses mainly the deployment of computational power and storage (Weinhardt et al., 2009). The next layer is called Platform as a Service (PaaS). It provides development platforms which build on IaaS in order to develop software applications. PaaS enables to develop, test, deploy, host, and maintain an application in an integrated environment. Often, the complete software lifecycle can be handled with PaaS (Weinhardt et al., 2009). The most common layer is Software as a Service (SaaS) which typically provides the customer interface (Weinhardt et al., 2009). SaaS applications operate on the hardware of the underlying IaaS layer and can be developed with the help of PaaS. SaaS is split up in two subparts which are real software delivered as services and web services (Weinhardt et al., 2009).In this paper, the focus is on the SaaS layer.

Benefits and risks of Software as a Service

The customer benefits of a SaaS solution, which are shown in Table 1, can be split up in three main categories: cost benefits, increase of flexibility, and miscellaneous benefits. The first item in the cost benefits category implies that no huge start-up investment is necessary. In contrast, when setting up a new software server in the traditional way, the server, the installation, and the software licenses would have to be paid. Further, typical SaaS offers do not consist involve single payments, but rather of a pay as you go manner (Armbrust et al., 2010), whereby one pays only for actually used resources (Weinhardt et al., 2009). Hence, fix costs are transformed into variable costs, which increases the transparency of costs. Furthermore, there are no long-term contracts and liabilities. The first benefit mentioned in the category Flexibility is the high scalability in both ends. Hence, customers are able to scale their resources up or down automatically or on-demand without limitations. This leads to an improvement in agility. Firms are able to immediately halve, double or triple their storage capacity, performance or user accounts in order to align to new conditions (Vaquero et al., 2009). By outsourcing parts of the firms' IT, security and reliability issues are handed over to the SaaS provider (Thomas and Redmond, 2009), thus equalizing the need for IT security staff. With the help of Service Level Agreements (SLA), the required performance is specified and guaranteed. A further benefit is the access to up-to-date hardware and software without concerns about updates and maintenance problems (Choudhary, 2007), thus allowing firms to focus on core competencies. Cloud Computing enables a much higher workload of hardware which causes an increase in efficiency and is therefore conform to Green IT (Iacobelli, Olson and Merhout, 2010).

Similar to the benefits of SaaS, we classified the risks into two categories: restrictions and IT risks. Handing over IT security issues to the SaaS provider is associated with a loss of control over systems and data (Thomas et al., 2009). The customer is not able to intervene in case of trouble, what results in restrictions. These restrictions may arise on the firm side in form of compliance requirements, or on the legal side. Legal restrictions depend on the particular country and its data privacy and financial laws which specify which data and systems are allowed to be outsourced and which must be kept in-house (Clarke, 2010). Because of the outsourcing of systems and data, dependencies to the SaaS provider arise. Once having shifted the system to a SaaS provider, the customer is not able to easily change the provider because of a lack of standardized interfaces (Armbrust et al., 2010).

Benefits of SaaS	
Cost benefits	- Low startup investments
	- Pay as you go
	- Transformation in variable costs
	- No long-term liabilities
Flexibility	- High scalability
	- Increased agility
	- Access to up-to-date hard- and software
	- No care for updates and maintenance
Miscellaneous	- Green IT
Risks of SaaS	
Restrictions	- Compliance
	- Legal issues
	- Data privacy
IT	- Loss of control
	- Security
	- Provider Lock-in

Table 1: Benefits and Risks of SaaS

User-Designed SaaS

The concept of traditional SaaS describes standardized software offered as a service delivered by the internet. Most of the setup and maintenance is taken over by the provider. Only a little effort through configuration and maintenance remains with the customer in order to achieve a running system. So far, the initial setup, operation, and maintenance tasks had to be done by the IT administrator. Without the help of the IT-department, it was not possible for the end-user to run or modify business software. This leads to dependencies on the IT administrator and inefficiency in more complex processes.

The new concept of user-designed SaaS arises mainly from computer supported collaborative work (CSCW). The end-user is able to create and modify software through the combination and adaption of existing tools. In the traditional SaaS, this was done by the provider or the IT administrator. In user-designed SaaS, only the initial setup is done by the IT-department; further adaptions are handed over to the end-user. This allows the end-user to design and adjust the system according to his needs and tasks. But it also requires a shift of the conceptual tasks from the IT administrator to the end-user. This skill has to be established and trained. After a short initial setup, the end-user is able to create own IT systems consisting of tables, documents, or websites shared with other users.

A pioneer in user-designed SaaS is Google Apps. Through the high level of recognition of Google's tools, users are already familiar with their use. Therefore, the lack of conceptual knowledge of end-users in contrast to IT professionals is attenuated. Google Apps bundles its tools like Mail, Docs, Calendar, Sites etc. in a suite adjusted to customer needs. These tools are highly integrated and aligned. This provides the possibility to create new tools through combination and adjustment of single tools, like online guidance systems or information platforms.

RESEARCH METHODOLOGY AND SETTING

Action Research

The applied action research approach was mainly chosen because of two characteristics: Firstly, the primary target of action research is to solve practical problems and examine scientific phenomena simultaneously (Hult, 1980). In our research

project the automation of processes shall be implemented, whereby influence and control on the units of analysis is given. Furthermore, scientific interest arises from the theoretical investigation of the benefits and risks of user-designed SaaS. The software creation aspect leads to the second argument. Relating to Otto (2010), action research is very suitable in information systems research for implementing systems or evaluating artifacts. Hence, it is the researchers' responsibility to support the practitioner or the company in solving their problem (Baskerville and Wood-Harper, 1996). His involvement might thereby shift "from being in charge of the overall project success (e.g., as the project manager), to being responsible for only a specific and minor part of the project activities (e.g., by conducting autonomous activities without being involved in project management)" (Simonsen, 2009, S. 3).

According to Susman (1978), action research consists of a cyclic process of five phases as follows: diagnosing problem, planning action, taking action, evaluating consequences, and specifying learning and documentation. Diagnosing the problem comprises the problem identification and definition. In action planning, the researcher and the practitioners collaborate on finding alternatives and choosing the best-fitting solution for the desired future state (Baskerville and Wood-Harper, 1998). Afterwards, the chosen solution is implemented and the effects are evaluated regarding to the expectations. Finally, based on the evaluation results and the experiences, the researchers apply a theory in the specific scenario. The assignment of our research site to these phases will be shown in the following section.

Research Site and Cases

The research site is the school of Business and Economics of a German university where 5.000 students are matriculated. Two projects were implemented. Firstly, the outdated information and communication processes of the faculty administration were improved with the help of an IT-platform. Secondly, an IT-supported information desk was implemented. Both projects were established with SaaS solutions.

Case 1: Faculty Administration

The first case is located at the administration of a faculty at a German university, at which about ten case workers support the head of faculty administration. The employees are responsible for tasks encompassing preparation of meetings, administration of rooms and facilities, human resources, budgeting, and organization of teaching and research. The case objective was the improvement and automation of processes focusing on information and communication.

First, the status quo was analyzed. Most tasks were done manually or only with basic IT-support. Prior externally programmed software required many resources for development and adaptations. Hence, another approach was desired. Agile prototypes, easy usage, customizability, and independency from the IT-programmer were the main requirements on which the different solutions were evaluated.

Classical software products were soon excluded as options as the setup, configuration, and maintenance could not be handled by end-users and would require their completion by an IT-employee. Furthermore, on-premise software could not be adjusted by end-users. SaaS solutions were investigated and a bid involving Google Apps was accepted. Google competitors' software was not operational at the case start in October 2009 while Google products were already familiar to many end-users through prior private usage.

In the action taking phase, a service platform was implemented with Google Sites, calendars and documents. Every section of the faculty administration was represented and additional services like congress or direction information were offered. The target was to provide basic knowledge in order to reduce personal, time-consuming enquiries. The platform was introduced in an employee workshop in order to match end-user expectations and individual training needs.

After the introduction of the new system, it seemed like we have to deal with acceptance problems arising by a slow IT-system adoption. With the help of personal interviews, we realized that the employees were not afraid of the new system because they could design it by themselves. Rather, they were afraid of the extraordinary work load in addition to their daily work resulting from caring about the system. Hence, we had to convince them of the advantages of the new system and the long-term time savings due to fewer requests. Regarding the success of a former information platform project in a similar setting, we knew that such a service platform could cut daily queries by about one-third. Revealing this experience resulted in a higher user acceptance. The shift of the conceptual tasks to the end-user was a further challenge. Regular workshops were initiated in order to bring up new ideas for beneficial content or functions. This initiative had success in common areas on the platform, but not on the individual sections. Hence, technical or conceptual issues were treated in individual consultation hours. Based on these consultation hours, further improvements were achieved, for example a virtual index for easier navigation and a protected area for internal documents.

In order to externally evaluate the SaaS-enabled service platform, a survey-based pre-test was realized before carrying out the platform online. The pre-test showed that the newly created service platform was rated better in terms of benefit, usability, and effectiveness than the previous solution.

As a result we figured out that user-designed SaaS affords a higher acceptance of the software. The end-user is able to adjust the software to the processes but has to carry out the conceptual tasks of the IT-administrator. This requires establishing and training of conceptual skills. The assignment of the case stages (S1) to the action research cycle is shown in Figure 1.

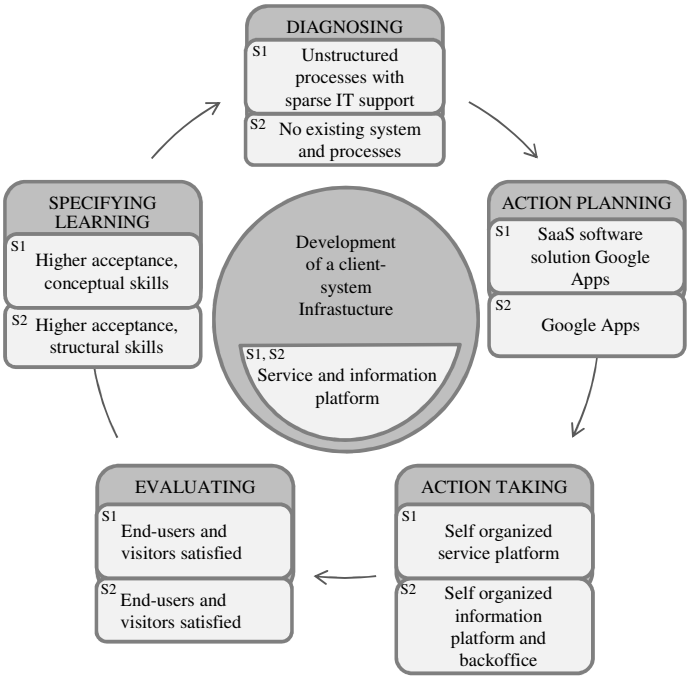

Figure 1: Alignment of the research cases to the Action research cycle

Case 2: Information Desk

In order to verify and extend the results from the primary case and to examine the effects more in depth, a further case (S2) was realized. It describes the setup of an information desk, including a physical room with several employees as well as an information platform, created and updated by these employees. In addition to an information platform like in S1, also the backend processes were implemented.

Diagnosing the initial situation resulted in the main tasks of this information platform: the guidance of students, employees and others, the further development of the website, compilation of statistics for transparency, and the constant scrutinizing of tasks.

Early in the action planning phase it was clear that such a project required IT support. It had to be easy to administrate and to use by the users because of the lack of IT-employees. The software had to provide functionality for cooperative creation of an information platform with integrated calendars, documents, and spread sheets. Finally, the tool should be free of charge and scalable. Because there was no legacy system existing, all processes had to be designed from scratch. Because of the positive experiences in Case 1, Google Apps was also deployed in this case.

In the action taking phase the first prototype of the platform was implemented. In order to perform all backend processes like time planning, request tracking, or task distribution, Google Docs was also deployed. If a request was sent to the physical information desk, it was tracked in a Google Docs spreadsheet, which was analyzed once a week. All requests which were not already posted on the platform were inserted in a wiki-system, administrated by the employees. In addition, the platform included nearly all information that could be interesting for students or employees. Hence, a calendar with all opening hours, e.g. of the library or the computer pools was integrated via a Google gadget. On the starting page, an online guide was

created. A Google Form led through a four-step questionnaire in order to answer a question, e.g. who is responsible for a specific task. Furthermore, an online job market for student workers was integrated. It enabled a central platform for chairs to publish their job advertisements and for students to search for jobs.

For evaluation, the implemented system, consisting of the information platform and the back office system, was considered. The results showed a high commitment and acceptance because employees were responsible for the system. In biweekly meetings problems were solved, tasks scrutinized and improvements discussed.

This case substantiated the thesis of the first scenario that user-designed SaaS increases the acceptance of IT systems. Here, the usage of the system was no additional task to their daily work; it was the core task of employees. That resulted in feeling more responsible for the self-created and -organized IT system. The shift of conceptual work to the end-users was observable, but led to a new problem. More IT skilled end-users recognized the advantages, but had no experiences with the conceptualization of IT systems. This resulted in an unclear structure and a bad organization of the IT system. Rules had to be defined in order to guarantee a more efficient structure. So the dependencies on the IT administrator were eliminated but experiences with the IT systems structure were required.

BENEFITS AND RISKS OF USER-DESIGNED SAAS

By applying the new user-designed SaaS approach in two projects, the benefits and risks were compared with the common benefits and risks of traditional SaaS. Common benefits and risks, which do not occur in the new model, are shown in Figure 2 in the upper left box. In the lower left box, the benefits which occurred in both types are listed, whereas in the lower right box the new arising benefits and risks were demonstrated.

Benefits

Common benefits that were not observable in user-designed SaaS were “Pay as you go” and “Transform fix costs in variable costs”. This is due to the fact that the education version of the software was used, which was free of charge. In the charged premium version, we suppose that these benefits would occur in a business setting. Furthermore, scalability was not observed, because we used not more than 50 users at one time. We were again not able to appreciate this issue. Regarding the Service Level Agreements (SLA) of Google, we assume that scalability is given in bigger projects.

“Less startup investments” and “No long-term liabilities” were observed in both types of SaaS. In our projects, the focus was not on these benefits because of the free of charge education version, but we were not bound to contracts nor did we invest money in hardware or software at the startup phase. We were able to observe an increased agility because nearly every modification could be implemented immediately by the end-user if they possessed the required conceptual knowledge. We could also identify “Access to up-to-date hard and software” as a benefit, because every update of the software was immediately online and accessible. Because the hardware and software was hosted at the provider side, we did not have to take care of updates and maintenance and were able to focus on our core competencies. The “Green IT” value could be observed because we did not have to buy servers or more powerful clients, but we were able to use the already existing hardware.

As shown in Figure 2, two new arising benefits could be observed. The first is the high user acceptance for a self-created IT system. The end-user is able to create and modify the IT-system by himself; that leads to a feeling of responsibility and to a higher efficiency. It can be traced to the fact that the user can design the system the way he needs it and the possibility to adapt it to arising requirements in the best way. The second benefit is the independency from the IT department. When the basic core system is configured, the end-user is able to do the conceptual work himself. This requires training of the required skills and control of the system efficiency. Therefore a trainer with conceptual skills and knowledge about IT systems is necessary.

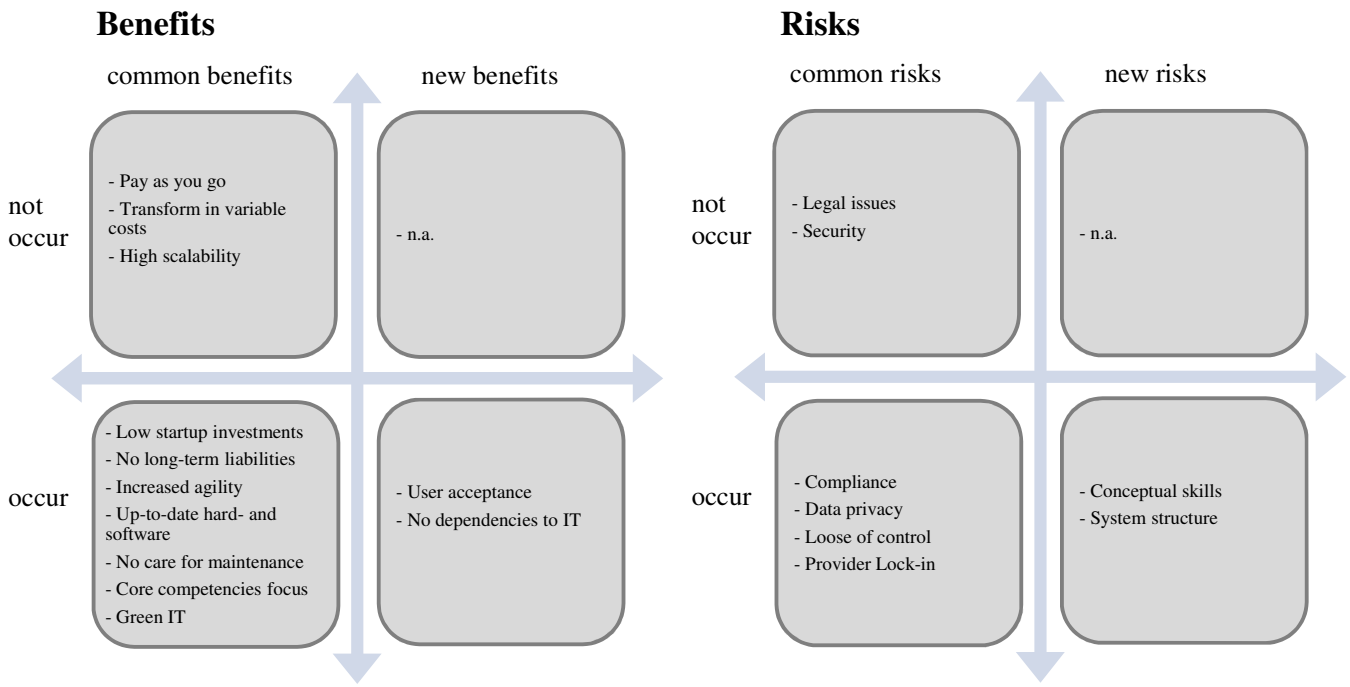

Figure 2: Benefits and risks comparison from the perspective of user-designed SaaS

Risks

In analogy to the benefits, we compared the risks of administrator designed- and user-designed SaaS. We were not able to observe "Security" and "Legal issues" risks. Security issues were handed over to the cloud computing provider. Hence, we had no influence on the security issues. Because we did not use security critical data, we were not able to evaluate the security risk within our research projects. Legal issues, like how to handle financial data, did not arise, so we are not able to make a statement about them either.

Risks that were observable in both types were "Compliance", "Data privacy", "Loss of control" and "Provider Lock-in". "Compliance" was given in terms of university restrictions. We were not allowed to integrate the platform in the university systems and domains because cloud computing was not allowed. To cope with this, we set up a domain especially for this project with a hyperlink from the university systems to the platform. A bigger problem was data privacy. Germany has very strict data privacy laws. Hence, the data collector is responsible for the collected data and the location of storage that has to be in the European Economic Area. Only a few exceptions allow the outer storage, for example in the United States. Furthermore, the data collector has to control the data. Those arguments are contrary to the cloud computing characteristics which declare that these tasks are handed over to the provider. In order to cope with these restrictions, we abandoned private data and focused on public information and communication processes. Furthermore, the loss of control could be observed. We did not know where our data is stored exactly and who had access to it. Hence, we had to trust the contracts and SLAs of Google. When we decided to use Google, we were bound to Google because of a lack of standardized interfaces. So it is a high effort to change the provider.

Still we were able to detect two unknown risks of user-designed SaaS. As mentioned already, the conceptual tasks are handed over from the IT department to the end-user. This requires the end-user to acquire conceptual skills. Usually, a non-IT employee does not possess such skills. Conceptual skills must be developed by an employee who is familiar with conceptualization and IT systems. In the beginning that employee has to monitor the efficiency of the newly created IT system. He has to control the structure and organization of the IT system as end-users do not have any experience in creating IT systems. Once realizing the benefit of such a system, end-users create content and functionalities without any structure behind it. That leads to inefficiency because of information overload and the lack of structure.

DISCUSSION AND OUTLOOK

User-designed SaaS provides the possibility for end-users to create software tools, especially in the field of CSCW. This requires the user to take on conceptual tasks which were so far carried out by IT administrators. Therefore, our research

interest focused on the additional benefits and risks brought up by new, user-designed SaaS solutions as compared to traditional, administrator-centered ones.

All in all, we figured out that many benefits and risks occur in both approaches with varying intensity. We were not able to observe the benefits "Pay as you go", "Transform fix costs in variable costs", due to the fact that we used a free version of Google Apps for education. Furthermore, we could not achieve high scalability because of the limited number of users on the examined platforms. By contrast, we were able to observe two by now unknown benefits in the field of user-designed SaaS. On the one hand, we experienced a very good user acceptance of the newly introduced system. This might be explained by the fact that users established and modified the software by themselves. Furthermore, we were able to reduce dependencies on the IT department. IT-administrators were only involved in the basic setup process. All conceptual tasks were handed over to the end-user.

In analogy to the benefits, most risks were observable in both approaches. Legal issues did not arise because we did not use critical data like financial or strategic information. Because we used only uncritical, public data, security was not a key issue. Our main interest was in the new arising risks in the user-designed SaaS concept. The first new risk resulted from the transfer of conceptual tasks to the end-user. Frequently, users do not have the skills to conceptualize and design software that solves their problems. This requires the end-user to establish and train the required skills. Hence, actions like workshops or consultation hours had to be provided. The second risk refers to the organization and structure of an IT system. The possibility to solve problems by creating own software tools might lead to an unstructured and uncontrolled growth of tools as well as data redundancies. Hence, it is not feasible to completely abandon the support of the IT department from the beginning, but it rather shifts to a person which trains conceptual skills and takes care of structured and organized IT systems. This support is necessary until the employees can handle the tasks on their own. The key aspect, the transfer of conceptual tasks to end-users, leads on the one hand to advantages like increased agility and flexibility and a higher system acceptance; on the other hand, uncontrolled growth of the system might occur and conceptual skills have to be trained.

As limitations we have to reveal that we only examined two cases in a quite similar setting, i.e., CSCW environments. Future research should examine the application in other contexts than CSCW as well as the development of management actions for facilitating the shift of conceptual tasks to end-users. In summary, user-designed SaaS seems to have great potential due to increased acceptance of IT systems and fewer dependencies to the IT department, and opens up new possibilities for designing and using software.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., et al. (2010) A View of Cloud Computing, *Communications of the ACM*, pp. 50-58.

Baskerville, R. L., & Wood-Harper, A. T. (1996). A Critical Perspective on Action Research as a Method for, *Journal of Information Technology, 11, 3*, pp. 235-246.

Baskerville, R., & Wood-Harper, A. (1998) Diversity in information systems action research methods, *European Journal of Information Systems*, p. 96.

BITKOM. (2009) *Cloud Computing - Evolution in der Technik, Revolution im Business* .

Choudhary, V. (2007) Comparison of Software Quality Under Perpetual Licensing and Software as a Service, *Journal of Management Information Systems*, pp. 141–165.

Clarke, R. (2010) Computing Clouds on the Horizon? Benefits and Risks from the User's Perspective, *BLED 2010 Proceedings*, pp. 569-590.

Gartner Inc. (2010) *Gartner's 2010 Hype Cycle Special Report Evaluates Maturity of 1,800 Technologies.*

Hult, M. a.-A. (1980) Towards a Definition of Action Research: A Note and a Bibliography, *Journal of Management Studies, 17, 2*, pp. 241-250.

Iacobelli, L. B., Olson, R. A., & Merhout, J. W. (2010) Green/Sustainable IT/IS: Concepts and Cases, *AMCIS 2010 Proceedings.*

Lawton, G. (2008) Moving the OS to the web, pp. 16-19.

Otto, B. (2010) IT Governance and Organizational Transformation: Findings From an Action Research Study, *AMCIS 2010 Proceedings*.

Simonsen, J. (2009) A Concern for Engaged Scholarship: The challenges for action research projects, *Scandinavian Journal of Information Systems*, p. 5.

Software & Information Industry Association. (2001) *Software as a Service: Strategic Backgrounder,* Software & Information Industry Association.

Susman, G. I. (1978) An Assessment of the Scientific Merits of Action Research, *Administrative Science Quarterly*, pp. 582-603.

Thomas, M. A., & Redmond, R. T. (2009) From the Client-Server Architecture to the Information Service Architecture, *AMCIS 2009 Proceedings.*

Vaquero, L., Rodero-Merino, L., Caceres, J., & Lindner, M. (2009) A Break in the Clouds: Towards a Cloud, *SIGCOMM Computer Communication Review*, pp. 50-55.

Wainewright, P. (2009) *zdnet.com*. Retrieved 01 17, 2011, from SaaS is surging in the downturn, says IDC: http://www.zdnet.com/blog/saas/saas-is-surging-in-the-downturn-says-idc/644

Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W., et al. (2009) Cloud-Computing – Eine Abgrenzung, Geschäftsmodelle und Forschungsgebiete, *Wirtschaftsinformatik*, pp. 453-462.