# User Generated Services during Software Introductions

Martin Schymanietz
University of Erlangen-Nuremberg
Lange Gasse 20, 90403 Nuremberg
Tel.: +49 (0) 911 5302-96398
martin.schymanietz@fau.de

Nivedita Agarwal
University of Erlangen-Nuremberg
Lange Gasse 20, 90403 Nuremberg
Tel.: +49 (0) 911 5302 - 881
nivedita.agarwal@fau.de

## ABSTRACT

In this paper, we describe the lack of user participation and involvment during software introductions. Especially big projects with a volume larger than 10 million US-$ are very likely to neglect important benchmarks like e.g. the budget or even completely fail. To fight these costly failures and support software introductions, we propose a service system that integrates the user into the software rollout. This service system consists of three service modules that are supported by components for feedback, communication, user incentives and motivation as well as. The service modules shall empower the users to give support and deliver tutorials or training to other users and furthermore establish a project specific platform which encourages a continuous improvement of the current software solution.

## CCS Concepts

**• Software and its engineering → Software post-development issues → Software evolution.**

## Keywords

Software introduction; service systems engineering; open innovation; user participation; user involvement

## 1. INTRODUCTION

Even if the percentage of successful software projects increased over the past 10 years, solely less than 40 % do not suffer from time/cost exceedances or a lower range of functions than originally intended. Approximately 20 % are even terminated before finalization or are not used after completion and therefore failed [1]. It is remarkable that especially large projects – having a labor content greater than 10 million US-$ - are only in 10 % of the cases successful while small projects (labor content < 1 million US-$) can reach a success rate of 76 % [1]. A Gartner survey about IT-Projects in North America and Europe showed a similar tendency with a failure rate of 28% at large projects (here: > 1 million US-$) and just 20% at small ones (< 350.000 US-$), resulting in both cases from the same reasons as above (functionality, time and cost overruns) [17]. Two of the main aspects that influence a software systems success are user participation and involvement. While the first term refers to users playing a vital role during the software Development, the second one emphasizes the importance and relevance of the project to the user [2]. A possible reason of the larger success of small software projects is their shorter duration and clearer definition of the project goals. This increases the communication between the relevant stakeholders and has a positive effect on user involvement [1]. Especially big IT projects do not only have an influence on current performance, but can trigger organizational change as well. In the following, we will focus on certain IT projects that do not only have a focus on technology, but on organizational performance as well. The main aspects, jeopardizing the technochange process are non- or misuse of the new technology or the usage without the expected benefit capturing [15]. To support the usage of a an IT system, currently user participation and involvement play an important role throughout the first phases of planning and testing of a software project [16], but not during later ones, i.e. during or after the software roll-out.

To promote user participation and involvement during software introductions and to achieve a higher success rate of software projects and actual usage, we take a service-systems-approach in which users are considered as so-called operant resources [3][4] who contribute actively to value creation in the system. Acknowledging the fact that users have so far played a passive role in the software introduction process we hypothesize that users so far tend to play mostly a passive role in the systems integration process and that additional efforts to encourage users to actively participate in the software introduction process can have a beneficial effect on the project success. This paper therefore studies the status quo in the practice of software introductions on a purposive sample of case studies. Our research interest is directed at the level of user involvement and participation, its explanation and the possibilities to increase it.

To reach an increased software usage by user involvement and participation, we suppose an IT based modular kit – we want to pilot in cooperation with industrial partners – containing service modules for: (1) user based support, (2) micro-learning and (3) collaborative software advancement. The first service module has as a goal to facilitate the search for company internal knowledge and to deliver support services. Beside of the essential pairing of knowledge seekers and providers, this module wants to lower the thresholds for an exchange of knowledge inside the company. In this context, all employees shall form a crowd, technically based on an internal platform. Second, the micro-learning module that consists – in an extreme case – of only one element like a single power point slide or videos shall replace extensive trainings and lead to a problem-orientated approach. However, the creation of a micro-learning unit will be based on a common concept, technical base and template. This will ensure a consistent look to enable the use of multiple consecutive units in one session without any media

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bearthis notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Requestpermissions from Permissions@acm.org.
OpenSym '16, August 17-19, 2016, Berlin, Germany © 2016 ACM. ISBN 978-1-4503-4451-7/16/08...$15.00
DOI: http://dx.doi.org/10.1145/2957792.2957812

disruptions. Third, the objective of the final service module is the continuous improvement of the introduced software. Ideas for certain enhancements will be collected on a common platform, where they can be rated and prioritized to realize relevant suggestions with a high utility first. This process will not be based on highly formal processes, but collect the real user experiences. Especially still untapped user resources shall become usable for user generated services through time allocations for further potentials of improvement. These three service modules will be further supported by suitable solutions for feedback, communication and user motivation. The solutions for feedback and communication may contain recommendations for technical solutions to use with the service modules like e.g. enterprise content management (ECM) systems, wikis, blogs, groupware- and community systems or social networks and help to analyze as well as to influence communication processes. To stimulate user participation, motivational and incentive aspects will be considered to ensure a proper participation in the creation of user generated services. Possible approaches can be the awarding of outstanding ideas and users, rating-systems or material incentives.

# 2. THEORETICAL BACKGROUND

## 2.1 Service Systems Engineering

In the past years, service research put a focus on service engineering. This approach contends that services can be developed in a nearly similar way as physical goods or software and shall reduce development times and costs as well as increase the quality of the offered services [5]. To reach these goals, service engineering delivers methods, procedure models and tools for a systematic development and design of services [6]. Service systems engineering (SSE) proposes to extend the current concept with design knowledge that is based on the creation, evaluation and execution of really existing service systems [7][8]. In other words, service systems engineering develops and designs service systems systematically and its goals are to support the design of evidence-based value creation in a contextualized and collaborative way. Service systems engineering has three research challenges: (1) engineering service architectures (2) engineering service systems interactions and (3) engineering resource mobilization. While the first term refers to the creation of new business models and cyber-physical systems with the aid of advanced tools and methods, the second one wants to improve collaboration under the usage of information systems. Finally, the third term claims to mobilize resources with information systems [7]. These challenges shall enable innovative value propositions [7] and in our case we will focus on engineering resource mobilization.

### *2.1.1 Engineering Resource Mobilization*

The technological advance, particularly in the field of information and communication technology (ICT), enables unprecedented possibilities for the interactive value creation within service systems. ICT enables mobilization of resources through mobilizing (1) human resources, (2) physical resources and (3) information resources [7]. While the first point refers to micro tasking [9] and/or service portals that promote the interaction between knowledge seekers and providers [7], the second one denotes resource sharing portals for physical private properties [7]. Examples would be sharing platforms like e.g. Airbnb, Uber or Zipcar. Finally, the last point relates to open data and/or user generated content [7][10][11]. In particular, these three aspects are seen as the focal points for the successful introduction, acceptance and active usage of new service systems within organizations and markets [7] and enable the involvement of users and customers into the innovation process [12]. Especially during software introductions, we see a strong need for resource mobilization that encourages the user to actively participate in the whole process to lead it to a success.

### *2.1.2 Service Systems Engineering & Open Innovation*

According to Reichwald & Piller 2009, open innovation can be applied to each of the five phases – (1) Ideation, (2) Concept Development, (3) Prototype-creation, (4) Product- and Market-Tests, and (5) Market Launch – of an ideal innovation process [13]. In this study we will focus particularly on the Product- and Market-Test phase. During this phase, companies can benefit from user feedback on functionality and troubleshooting. This plays an important role especially when there is insufficient user participation in the earlier phases. Even if the terminology fits better into our focus, the phase Market Launch is marketing centered and describes actions that promote the introduction to a new market like e.g. distribution channels or pricing mechanisms [13]. Therefore, from our viewpoint the service system that establishes user generated services during software introductions can be seen as an extended product- and market-test that integrates the foundations of open innovation.

## 2.2 Technochange

In 2004, Markus coined the term technochange for the creation of "high-risk, potentially high-reward, situations" enabled through the use of IT to "trigger major organizational changes" [15]. The technochange life cycle can be divided into four phases: (1) Chartering, (2) Project, (3) Shakedown,(4) Benefit Capture. While the first phase refers to the initial step where the technology oriented change is planned, authorized and funded, the second one covers the solution development where the technology is purchased or constructed. This phase ends with the software release. During the shakedown, the organization starts to use the software and wants to reach a normal operation. Finally, the benefit capture phase describes a systematic approach for deriving benefits from new processes or similar. One main aspect observed during technochange projects are exported problems. These are problems coming up in one phase of the technochange lifecycle, but are not fixed or detected. These problems can have future repercussions on the success of the following phases, due to e.g. unavailability of resources or budget [15].

# 3. RESEARCH DESIGN

The following research regards software introductions as a service system that contains not only developers as active participants, but users as well. In this context, we want to answer the following two questions:

1. How is the flow of information between the participants in such systems formally designed?
2. What kind of contributions are generated by the users?

Furthermore we want to enrich the current research on software introductions with the principles of SSE – especially with the aspects from resource mobilization like micro learning and user generated content.

This study is research in progress, has an explorative nature and does not claim completeness. Rather, it wants to collect with a purposive sample of different software introduction projects the variety of possibilities for user participation- To assess the deficits in user involvement and participation during the introduction of new software, we conducted nine semi-structured interviews with employees from an IT service provider. These interviews were evaluated under the problem centered interview approach. The following table contains the job descriptions of the interviewees as well as the branch of the regarded technochange projects.

**Table 1. Job Descriptions of the Interview (IV) Partners**

| # | Job Description | Branch |
|---|---|---|
| IV 1 | Key Account & Business Development Manager | Automotive |
| IV 2 | IT Consultant Requirement Engineering | Arts |
| IV 3 | IT Consultant | Automotive |
| IV 4 | Software Developer | Logistics |
| IV 5 | Project and Change Manager | Services |
| IV 6 | IT Consultant Requirement Engineering | IT |
| IV 7 | IT Consultant | Automotive |
| IV 8 | IT Consultant Software Development | Education |
| IV9 | Test Manager | Federal Agency |

## 3.1 Problem Centered Interviews

Problem centered interviews “aim to gather objective evidence on human behavior as well as on subjective perceptions and ways of processing social reality” [14]. It contains three basic principles: (1) A problem centered orientation on relevant social problems, (2) methodical versatility through an object orientation and (3) a process orientation that leads to an open interview partner which feels appreciated. The evaluation is based on transcribed interviews and integrates a case study with a typology that enables cross links between both, the various parts of a single interview and across the single cases [14]. The interviews followed a semi-structured guideline that asked for a technochange project with a focus on the shakedown and benefit capture phases and how user participation took place during this project phase.

# 4. FINDINGS

The initial interviews showed that most of the regarded software introductions lack proper end-user participation and involvement. If user participation and involvement were executed, the group was limited to selected key users who were able to give feedback during the software rollout – but excluded the large amount of end users [IV1][IV3][IV7] [IV9]. The feedback had a unidirectional direction from the lead-users to the project team. Furthermore, the lead-users could not officially communicate with each other to search for problem solutions – except of an informal exchange of information at meetings or based on personal contacts [IV1][IV3][IV5][IV7]. A reason for the missing involvement was that an extensive user participation was not planned in the projects [IV1][IV3][IV4] – except of some key user interviews [IV3][IV7], workshops [IV8] or trainings [IV1][IV3][IV5]. The non-integration of the final users was in particular based on time and budget constraints [IV1] [IV7][IV8], bureaucracy [IV5][IV6][IV7] or the company’s philosophy [IV1][IV3][IV4][IV5]. Especially hierarchical structures, where one or maybe a couple of executives decide about the procedure, prevent companies from a deeper user involvement [IV1][IV3][IV4][IV7][IV9].

However, not all projects suffered from this lack of user participation. Some projects had a more interactive setting and enabled a bidirectional communication between users through e.g. social media [IV2][IV8]. For example, the users were seen as a part of the whole and that the project would not become a success without their participation [IV2] and user involvement is a valuable commodity that increases the acceptance of the entire project [IV9]. But even if there was a possibility for users to participate, it could be observed that the users need to be motivated for a proper participation that supports the whole software introduction [IV2]. IV4, IV7, IV8 and IV9 stated that from their point of view, a deeper participation with feedback possibilities – especially of experienced key users – is very beneficial for the project success [IV4] [IV7][IV8][IV9], but was not inside the scope and budget of the project and therefore impeded the project team from a deeper user participation due to the overhead that comes with it [IV7]. In this context, IV9’s successful project (in time/budget; >75.000 users) contained an e-learning platform, a wiki, a knowledge database and a ticketing system for user feedback [IV9]. Additionally, piloting phases where the users should have enough time for usability tests and search for errors or bugs that become apparent not before usage, seem to be a good first step to a better user involvement [IV6][IV7][IV8][IV9]. Furthermore, IV5, IV8 and IV6 remarked that communication between the project team and the end users is a key factor that influences the project outcome [IV5][IV6][IV8] – independent of the technical solution used [IV5]. To ensure a common language for communication and understanding of what is demanded – especially when expert knowledge is missing – the establishment of a glossary is beneficial [IV6]. For IV8, user generated services would be an ideal solution to achieve the project goals and a proper use of the implemented software solution in the long run [IV8]. However, user generated services only make sense, if they are well-kept and used [IV5].

Another aspect is the need for a proper support during and after the software introduction. Even years after the software roll-out, some users come back to the developers and ask for support [IV1, IV2]. However, the arising questions have often been already clarified, but were not easily detectable for the person looking for support [IV1]. IV7 stated that the expected amount of improvement suggestions from users prevent the project team from implementing such a possibility [IV7]. But even if a user help desk was established, this does not automatically imply that they can help the users properly. These help desks are often outsourced and lack of an appropriate understanding of the introduced software. In the end, these deficiencies are very likely to lead to acceptance problems among the daily software users [IV9].

# 5. PERSPECTIVES

From the conducted interviews, three main problems of software introduction processes aroused that we want to approach with our service system: (1) the deficits in participation and involvement of a broad user base during the software roll-out, (2) their unidirectional communication and (3) possibilities to give improvement suggestions are neglected due to an expected overhead. In this context, we assume that our approach of an open service system where all users are able to give support, create learn units as well as have the possibility to introduce new ideas or suggest improvements has the ability to lead to a better general usage of an introduced software through an activation of former passive users. Furthermore, the direction of interaction among the small amount of active participants can be extended to a bidirectional exchange of a larger group of users by the three service modules – especially in larger projects with a broad user base. Motivational aspects to ensure a proper and frequent usage of both the new software as well as our proposed service system will be covered by our second module about motivation and user incentives. Finally, we think that our service system has the potential to reduce the overhead in the support departments in two ways: On the one hand, some tasks can be outsourced to users having the right expertise. This includes the creation of learn units that have the potential to reduce the necessity of trainings as well as the solution of current user problems with the help of the crowd. On the other hand, suggestions for improvement that emerge after the roll-out do not need to be screened by the staff, but shall be selected by the whole community to derive relevant ideas – carried out by a user driven voting and prioritization process.

The next steps in our research about our proposed service system – that enables all users of a software to improve and shape it – are the following: After the initial development of the single modules, these will be piloted in cooperation with our industrial partners – accompanied by additional validating interviews. After this first phase of piloting, an iterative process shall improve and then connect all of the modules to conduct a second testing phase. During the second piloting the whole service system will be implemented at our partners, giving us the possibility to examine how the five elements are influencing each other. The results shall enable us to finalize the modules and transfer them to the practice and public – among others through massive open online courses (MOOCs).

# ACKNOWLEDGEMENTS

The research presented in this paper was partially funded by the German Federal Ministry of Education and Research in course of the project ExTEND (http://projekt-extend.de/), FKZ 01FJ15127.

# REFERENCES

[1] Standish Group International 2013. *The Chaos Manifesto.*

[2] Abelein, U., Sharp, H. and Paech, B. 2013. *Does Involving Users in Software Development Really Influence System Success?.* In *IEEE Software,, Vol. 30, Issue 6, p.17-23.*

[3] Vargo, S. and Lusch, R. 2004. *Evolving to a New Dominant Logic for Marketing*. In *Journal of Marketing*: January 2004, Vol. 68, No. 1, pp. 1-17.

[4] Vargo, S. and Lusch, R. 2008. *Service-dominant logic: continuing the evolution*. In *Journal of the Academy of Marketing Science*. March 2008, Volume 36, Issue 1, pp 1-10.

[5] Bullinger, H.-J. and Scheer, A.-W. (ed.) 2003. *Service Engineering – Entwicklung und Gestaltung innovativer Dienstleistungen*. Berlin: Springer Verlag.

[6] Bullinger, H.-J. 1999. *Entwicklung innovativer Dienstleistungen*. In H.-J. Bullinger (ed.), *Dienstleistungen – Innovation für Wachstum und Beschäftigung*. Wiesbaden: Gabler.

[7] Böhmann, T., Leimeister, J. M. and Möslein, K. 2014. *Service Systems Engineering - A Field for Future Information Systems Research*. In *Business & Information Systems Engineering (BISE)*, April 2014, Volume 6, Issue 2, pp 73-79.

[8] Satzger, G., Ganz, W., Beck, R., Benkenstein, M., Bichler, M., Bienzeisler, B., Böhmann, T., Dunkel, W., Fähnrich, K.P., Gemünden, H.G., Gouthier, M., Gudergan, G., Kieninger, A., Kölling, M., Krcmar, H., Kremer, A., Möslein, K., Schultz, C., Stauss, B., Stich, V., Weinhardt, C. 2010. *Auf dem Weg zu einer Service Science – Perspektiven, Forschungsthemen und Handlungsempfehlungen aus der Sicht einer interdisziplinären Arbeitsgruppe. Empfehlungen an die Taskforce Dienstleistungen im Rahmen der Forschungsunion Wirtschaft-Wissenschaft*. Arbeitsgruppe "Evaluation Service Science" der Taskforce Dienstleistungen.

[9] Kern, R., Bauer, C., Thies, H., Satzger, G. 2010. *Validating results of human-based electronic services leveraging multiple reviewers*. In: *Proc Americas conference on information systems, Lima.*

[10] Lindman J., Rossi, M. and Tuunainen, V.K. 2013. *Open data services: research agenda*. In *Proc 46th Hawaii international conference on system sciences (HICSS), Maui.*

[11] Leimeister, J.M., Huber, M., Bretschneider, U., Krcmar, H. 2009. *Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition.* In *Journal of Management Information Systems, 26(1):197-224.*

[12] Magnusson P. 2003. *Benefits of involving users in service innovation.* In *European Journal of Innovation Management 6:228-238.*

[13] Reichwald, R. & Piller, F. 2009. *Interaktive Wertschöpfung*. Wiesbaden: Gabler Verlag.

[14] *Witzel, A. 2000. Das problemzentrierte Interview. In Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 1(1), Art. 22, 2000.*

[15] Markus, M. L. (2004). Technochange management: Using IT to drive organizational change. *Journal of Information Technology*, *19*(1), 4–20.

[16] Semmann, M., & Böhmann, T. (2015). Post-Project Benefits Management in Large Organizations – Insights of a Qualitative Study. Thirty Sixth International Conference on Information Systems (ICIS), (June), 1–16.

[17] Mieritz, L. (2012). Gartner Survey: Why Projects Fail. Published: 1 June 2012. ID:G00231952.