Designing an End User Participation and Involvement Assistant for Continuous IS Development

Research-in-Progress

Phillip Haake
University of Mannheim and
Karlsruhe Institute of Technology
(KIT), Germany
haake@es.uni-mannheim.de

Stefan Morana
Karlsruhe Institute of Technology
(KIT), Germany
stefan.morana@kit.edu

Silvia Schacht
Karlsruhe Institute of Technology
(KIT), Germany
silvia.schacht@kit.edu

Lu Zhou-Hartmann
University of Mannheim, Germany
zhouhartmann.lu@gmail.com

Alexander Maedche
Karlsruhe Institute of Technology (KIT), Germany
alexander.maedche@kit.edu

Abstract

The nature of information system provision has changed as the business models of software firms are continually changing towards software-as-a service and platform-as-a-service solutions. Involving end users in a continuous development of such information systems promises to increase the overall system's success. The conceptualization and actual realization of the user involvement concept is subject of ongoing research. Based on a systematic literature review, we identified three implications of this research that are relevant for the development of an information technology based assistant for user participation and involvement. In this paper, we introduce our Design Science Research project on the theory grounded design of a user participation and involvement assistant. We aim to enable end users to provide meaningful feedback to the developers during their actual system use. Moreover, the assistant will enable developers to effectively analyze the vast amount of complex end user feedback.

Keywords: Issue tracking system, Assistant system, User involvement/participation

Introduction

Nowadays, information systems (IS) are ubiquitous in private life and organizations. For instance, businesses employ IS for improving their operations and their value chains (Davenport 1998; Mabert et al. 2001). Nonetheless, the nature of IS provision has changed in the past as the business models of software firms are continually changing towards Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS) solutions. This trend has brought new challenges to software development for software firms as they can potentially trace and access end users of their IS while it is in use. It also allows end users in organizations to interact directly with the software firm on matters regarding their particular piece of software in use. New ways of continuous IS development with user participation and involvement (UPI) are, therefore, possible based on current technology. For this context, we define users as the end users who use the system in their work practices to accomplish their work tasks. In addition to understanding user behavior, researchers are analyzing UPI in system development to ensure system success. It is widely acknowledged that UPI is a critical factor for system success (Cavaye 1995; Harris and Weistroffer 2009; Ives and Olson 1984). However, most of knowledge and research results are based on the initial phase of system adoption or system development. Many studies suggest to involve users in initial phases of IS development, such as defining task requirements and system design, which, in turn, will significantly and effectively influence the success of a system. One reason could be that systems designed by UPI capture users' needs more accurately (El Emam et al. 1996; Kujala 2008; Lu and Wang 1997; Yetton et al. 2000). One might add that user feedback can also be valuable for testing a software application (Foster and Franz 1999; Kappelman 1995; Kawalek and Wood-Harper 2002). However, the success of users' initial adoption is not equal to the continued success of an IS. The value of an adopted IS heavily depends on whether end users make use of it (Bhattacherjee and Sanford 2006) and how they use it (Bhattacherjee 2001; Karahanna et al. 1999). UPI in continued IS development in the post-adoption phase is rarely discussed in the current literature (Abelein and Paech 2013; Kim and Crowston 2011). Continuous development is important to ensure the main goal of continued use by end users and ultimately organizations (Carroll 2004; Hansson et al. 2004, 2006). Continuous development includes the cyclical iteration of continued design and user evaluation for achieving system performance improvements (Carroll 2004) and extends the traditional development life cycle to the use phase (Åsand and Mørch 2006; Fischer 2008; Lieberman et al. 2006). Thus, it is important to identify end users' often-changing requirements to adjust a running system to new and changing use contexts, while maintaining the underlying qualities of the particular IS in use, such as its usability. A comprehensive analysis of UPI and its potential for continuous IS development to ensure continued use is therefore warranted. As a result, we came up with our overall research question:

How does user participation and involvement affect continuous IS development?

The answer to this research question allows us to identify an area of particular need for an improved management of user-developer interaction. Ubiquity, quality, and accuracy of end user feedback are critical for developers' enhanced understanding of the particular usage context of end users and their evolving requirements. Particularly, the interaction of end users and developers in the ticketing process of an issue tracking system (Just et al. 2008) is a crucial part of involving users in the continuous IS development process, as users in the use phase are more likely to provide feedback (cf. Wagner and Piccoli 2007). Furthermore, it is important for developers that this feedback is meaningful (cf. Yetim et al. 2012) and easy to analyze (Pagano and Bruegge 2013). However, the information and requirements that software developers receive from users are generally incomplete due to a variety of cognitive, communicative, and motivational reasons. Thus, actions taken by developers that base on the incomplete information are often unsuccessful (Browne and Rogich 2001, p. 224). On the contrary, the usage of tool support for users' post-deployment participation shows first positive results with respect to usability and usefulness (Yetim et al. 2012). Inspired by this prototypical concept, we aim to investigate in detail how to support users in their formulation of requirements and grievances with a SaaS in the form of tickets. In this work, we therefore specifically aim to answer the following question:

How to design an assistant system that enables ticket generation in continuous IS development?

In our subsequent research, we aim to develop an assistant system ('UPI Assistant') for the ticketing process of an issue tracking system for a campus management software (CAMPSYS) at a Central European university with approx. 10.000 students, who are the users. We will derive and evaluate several design features of the UPI Assistant in a laboratory experiment and a related field experiment that will follow the initial evaluation of features in the laboratory. Consequently, we contribute to existing research by combining and evaluating several feedback mechanisms in experimental settings. Especially, the scientifically grounded combination of feedback mechanisms by following the design science approach represents the novelty of our research. This will improve the interaction between users and developers in the post-adoption phase.

Theoretical Background

For the case of many IS in the use phase, a large number of unknown end users is widely dispersed in different locations, organizations, use contexts, or work groups with different goals for the use of the IS. Moreover, people use IS differently on an individual level (DeSanctis and Poole 1994; Majchrzak et al. 2000; Orlikowski 2000). They may also use a technology for unanticipated purposes and at the same time their practices are shaped by the technology in use (Carroll 2004). On the basis of a management perspective, Orlikowski and Hofman (1996, p. 267) have referred these changes that arise from local innovation and are not anticipated or planned as ‘emergent changes’. These changes can arise during the implementation as well as the use phase of a technology in an organization. We aim to link this description of innovative and emergent usage with influences of user participation and involvement (UPI) identified in the literature. Researchers identified UPI as a factor, which can positively influence the perceived usefulness and ease of software use (Jasperson et al. 2005). Perceived usefulness of a new IS becomes less relevant in the post-adoption phase as users make their own actual usage experiences (Hsieh and Wang 2007; Karahanna et al. 1999) and they adapt their usage behavior accordingly (Jasperson et al. 2005). However, the confirmation of positive expectations towards an IS, the perceived usefulness, perceived ease of use, and actual performance in use influence users’ satisfaction and are thereby important indicators for the intention for continued use (Bhattacherjee and Lin 2014; Bhattacherjee 2001; Qin and Xu 2007; Wixom and Todd 2005). In addition, satisfied users are more likely to use a broader set of features or to explore the use of system features to meet emergent needs and can give feedback regarding these needs (Saeed and Abdinnour 2013).

A better understanding of the aforementioned constellation of influencing factors between usage and UPI is required. The terms user involvement and user participation were frequently used as the same concept (Harris and Weistroffer 2009) and were also often referred to as ’user engagement‘ (Hwang and Thorn 1999). Nevertheless, a distinction between these two terms is possible (Barki and Hartwick 1989). Barki and Hartwick (1989, p. 53) defined “user involvement as a subjective psychological state reflecting the importance and personal relevance of a system to the user” and user participation can, therefore, be measured as “a set of behaviors or activities performed by users in the systems development process” (Barki and Hartwick 1989, p. 53). Furthermore, in a later article, they found that initial user involvement has a positive effect on user’s attitude towards a new IS and that user participation affects subsequent levels of user involvement (Barki and Hartwick 1994). Participating users can also be seen as a subset of those users who are involved (Bano and Zowghi 2013). To establish a link between UPI and continuous IS development, we conducted a literature review of empirical studies published between 1990 and 2015. We decided to set the time limit on the year of publication, because we found that studies which focused on IS projects in the nascent stage of the field before 1990 show conflicting results (Cavaye 1995; Olson and Ives 1981). There are several potential reasons for these conflicting results. Olson and Ives (1981) as well as Cavaye (1995) attributed the conflicting results to the used research methods in the early days of the research stream. For instance, 80% of the identified studies for the period 1980-1989, which were analyzed by Bano and Zowghi (2015), conducted survey based research. As surveys can hardly capture complex concepts that are not fully understood, there is a large variance in early results (Cavaye, 1995; Bano and Zowghi, 2015). Another reason is the synonymous use of the terms of user involvement and user participation and other termini in many papers (Bano and Zowghi, 2015). This makes it difficult to attribute success or failure to a certain approach. Abelein and Paech (2013) add that some of the negative correlations can be explained by the context of the individual studies, such as the complexity of an implemented system. However, more current research indicates a positive relationship between user involvement, participation, and system success (Abelein and Paech 2013; Bano and Zowghi 2013; Harris and Weistroffer 2009; Hwang and Thorn 1999). We conducted the literature review by following the guidelines for performing a systematic literature review by Kitchenham and Charter (2007). To identify search terms and their commonly used synonyms, we reviewed two other related literature reviews on UPI (namely Abelein and Paech 2013, and Bano and Zowghi 2013). We developed our search string from their basis and chose the following databases for our review, as they were the predominant sources in previous reviews: Google Scholar, Science Direct, Springerlink, IEEE xplore, ACM DL. In our quality control, we focused on papers with detailed description of an empirical research method to ensure the relative accuracy of the obtained insights (Abelein and Paech 2013). In total, we retrieved 1,400 papers in the first application of the search string on the chosen databases. 1,362 of these studies were irrelevant, duplicates, or excluded after identification of exclusion criteria (Kitchenham and Charters 2007). We added a number of 25 studies through scanning the references of the remaining papers (backward search) and crosschecking them with the two earlier reviews (Abelein and Paech 2013; Bano and Zowghi 2013). As a result, we ended up with 63 relevant papers. A final number of 57 was left after the application of quality control criteria (Kitchenham and Charters 2007). Out of these 57 studies, 45 (79%) indicate a positive impact of user involvement on system continuance. This is in line with earlier results (Abelein and Paech 2013; Bano and Zowghi 2013). Only 14 articles (namely Begier (2010), Bellucci et al. (2015), Bragge and Merisalo-Rantanen (2009), Finck et al. (2004), Hansson et al. (2004, 2006), Hendry (2008), Kabbedijk et al. (2009), Pagano and Bruegge (2013), Pagano and Maalej (2013), Parmentier and Gandia (2013), Wagner and Piccoli (2007), Wilson et al. (2012), and Yetim et al. (2012)) focused on UPI in continuous IS development. All of them posit a positive relationship between UPI and the success of continuous IS development.

Initially, we listed all methods for UPI in continuous IS development based on the objectives linked to them, which we identified during the literature review. In line with our research question, we focused on the goal of obtaining feedback from end users. This approach is tied to the idea of enhancing the accuracy of requirement descriptions and exposing the limitations and failures of a current solution. It also includes obtaining information on suggestions for new features and other solutions. The employed methods for UPI are user meetings, usability workshops (Kabbedijk et al. 2009), courses for discussing the objectives of a new system introduction (Hansson et al. 2004, 2006), newsletters, customer relationship management tools (Hansson et al. 2006), mailing lists, bug trackers (Hendry 2008), analysis of discussion forums (Finck et al. 2004; Pagano and Maalej 2013), surveys (Begier 2010), user toolkits (Parmentier and Gandia 2013), and IT helpdesks (Bragge and Merisalo-Rantanen 2009; Hansson et al. 2004, 2006; Hendry 2008). Some authors also discuss methods such as web interfaces for feedback and proposals (Hansson et al. 2004, 2006), and feedback functions within an IS. Subsequently, we analyzed potential benefits and problems of UPI for continuous IS development to illustrate the gaps in the current level of insights in the UPI literature. According to Wagner and Piccoli (2007), the most valuable UPI for IS development takes place after the release of the developed system and the real engagement of a wider group of users. Due to their actual usage, users in the post-adoption phase are more motivated to engage, as they are able to contribute meaningful feedback based on their actual use experience. In all 14 studies on continuous IS development, the researchers mentioned that involving users in gathering further requirements is beneficial. For instance, as users' work practices might change during use of the technology, they may convey new requirements for a coming release (Pagano and Bruegge 2013). UPI in continuous development can also create a sense of co-operative development in users and lead to their increased engagement, a sense of empowerment, and ownership (Hansson et al. 2004, 2006). Furthermore, involving users as co-developers (Bellucci et al. 2015; Wilson et al. 2012; Yetim et al. 2012) allows extensive exploration and the experience of user appropriation scenarios (Bellucci et al. 2015). It also enables developers to benefit from users' personal innovativeness (Bellucci et al. 2015; Bragge and Merisalo-Rantanen 2009; Finck et al. 2004; Hansson et al. 2004, 2006; Kabbedijk et al. 2009; Parmentier and Gandia 2013; Wagner and Piccoli 2007; Wilson et al. 2012; Yetim et al. 2012). However, if UPI is not integrated in the development process appropriately, it can cause serious problems instead of benefits. Twelve out of 14 papers about continuous IS development discuss difficulties of UPI for system design in terms of problems with gathering users' feedback, such as high volume and poor quality of the feedback (cf. Pagano and Bruegge 2013). On the contrary, little feedback does not ease the analyzing work for developers. Bragge and Merisalo-Rantanen (2009) found that a reduced amount of feedback, gathered from unmotivated users, mostly contains improper or irrelevant information. Even without volume constrains, user's feedback is often unqualified in terms of misunderstanding the features, and general misuse and configuration of the software (Pagano and Bruegge 2013). Furthermore, users are often not willing to participate in helping developers with the analysis and evaluation of a very new system, since this involvement results in extra work load (Wagner and Piccoli 2007). When involving representatives of end users in the design phase, another challenge emerges, which is associated with representatives' personal interests that can become overvalued (Wilson et al. 2012). Other issues that might negatively influence the system design are the feedback transfer and interpretation between users and developers. Often, users articulate their feedback in their natural language and describe issues or requests in a particular manner. Developers have problems to understand what users aim to convey (Pagano and Bruegge 2013), which consequently decreases the quality of design.

The findings from our literature review indicate the following core implications as summarized in Table 1 that are relevant for the development of an assistant for UPI. We address these implications and aim to develop an assistant for UPI. Thus, we propose the design of a UPI assistant, which enables end users to provide feedback or requests addressing the system in the form of tickets (Just et al. 2008; Serrano and Ciordia 2008; Zhou et al. 2012) for an issue tracking system.

Table 1. Implications derived from literature

	Description	Sources
Implication 1	End users are more likely to be motivated as participants whenever they can provide feedback in their actual use context.	(Bragge and Merisalo-Rantanen 2009; Finck et al. 2004; Hansson et al. 2004, 2006; Kabbedijk et al. 2009; Pagano and Maalej 2013; Wagner and Piccoli 2007)
Implication 2	End users need to be enabled to provide meaningful feedback to developers during use.	(Bellucci et al. 2015; Pagano and Bruegge 2013; Yetim et al. 2012).
Implication 3	Developers need to be enabled to effectively analyze the vast amount of complex end user feedback.	(Hansson et al. 2004, 2006; Hendry 2008; Kabbedijk et al. 2009; Pagano and Bruegge 2013; Pagano and Maalej 2013; Yetim et al. 2012)

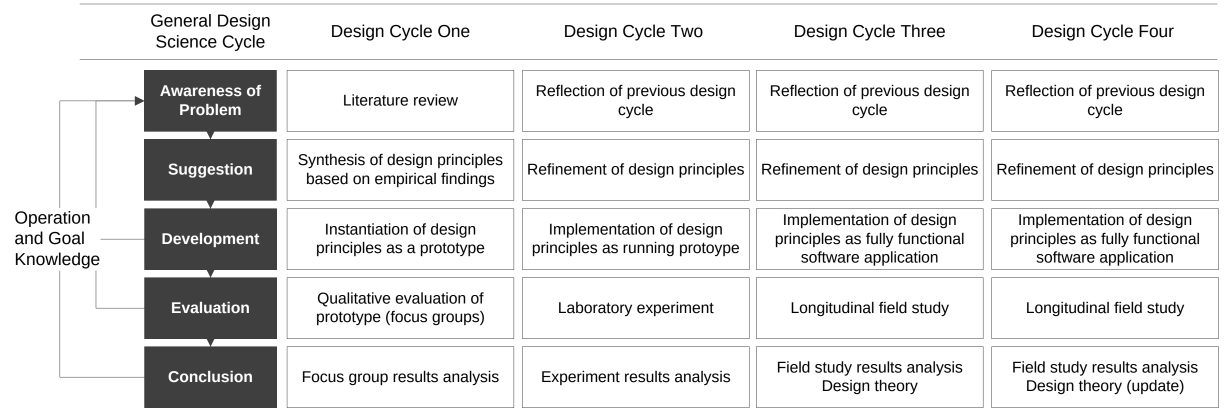
Design Science Research Project

In order to design a UPI assistant, which enables end users to articulate their requirements and feedback for their currently used IS, we follow the Design Science Research (DSR) (Hevner et al. 2004) approach by Kuechler and Vashnavi (2008). We argue that involving end users in the continuous IS development process enables increased IS's success. Existing feedback mechanisms, e.g. the option to provide textual feedback to developers in, for example, MS Office, or calling the service department of a software company, enable end users to participate in the development of the respective system. However, we argue that there is an opportunity to improve such feedback mechanisms by combining various techniques into an integrated "end user involvement" assistant. We argue that such an assistant, which combines various feedback and communication mechanisms, enables end users to provide feedback that is more meaningful. Thus, it enables end users to participate in the development of the system. Providing assistance to users is an ongoing research effort in IS research (Maedche et al. 2016) and related domains such as manufacturing (Yang and Plewe 2016). In addition, it is part of the overall effort of understanding and optimizing the human-machine interaction (Wandke 2005). Moreover, the resulting design will not only address the solution for our actual problem instance, but also result in a general design applicable for other, related problem instances. We aim to design a UPI assistant for end user involvement in continued IS development. Furthermore, we aim to understand the outcomes of using such an assistant with respect to end users, developers, and the overall system success.

We decided to design and implement the UPI assistant in the context of a university enabling us to test the design artifact easily with an appropriate number of users. However, when conducting studies with students, the question of the results' reliability arises with respect to their applicability and generalizability for the real world. In order to address these issues, we selected a campus management software (CAMPSYS), because it covers a system environment, in which students are the actual end users. Our research context and actual problem instance is the improvement of CAMPSYS at a large German university. CAMPSYS provides students the functionality to enroll for courses, register for examinations, get an overview on grades, etc. The university in question implements CAMPSYS to substitute a legacy system. The key decision makers of CAMPSYS believe that end user involvement will lead to increased system success. CAMPSYS is a standard software solution customized to the specific requirements of the university (our research site). The system is operated by the university's computing center. The software provider as well as the university's computing center continuously develop CAMPSYS. Thus, we argue that the selection of students and the CAMPSYS is appropriate for investigating continued IS development. Moreover, as we selected a standard software solution, which is customized, implemented, and continuously developed in an organization, we argue that the selected research context ensures the applicability and transferability of our results to other real world contexts. In addition, we ensure the incorporation of theory and existing design knowledge as well as practical insights and feedback in our research (Hevner 2007) by following the DSR approach.

Our research project consists of four consecutive design cycles as depicted in Figure 1. The **first design cycle** aims at the derivation of the design principles and their instantiation as an assistant for end user involvement in the form of a clickable mockup. First, we will raise problem awareness based on an extensive and rigorous literature review on continued IS development. Building on this, we will derive meta-requirements. Subsequently, we will derive design principles based on empirical findings and theories to address the two identified implications. Thus, we plan to integrate existing research addressing the design of feedback and communication mechanisms to inform the design. Next, the design principles will be instantiated in the form of a clickable prototype and qualitatively evaluated within focus group workshops (Tremblay et al. 2010). We intend to conduct focus groups with students (the end users) and developers of CAMPSYS. Due to the regional proximity of the software provider of CAMPSYS, we will also invite some developers of CAMPSYS to join the focus groups. Thus, we will receive feedback on the assistant's design from multiple perspectives. We will conclude the first design cycle with the analysis of our qualitative evaluation.

The **second design cycle** aims for the refinement of the design and the implementation of the assistant as a running software artifact. We will start the design cycle with the reflection of the previous design cycle and the potential refinement of our design principles based on the findings from the focus group workshops. Next, we will implement a running UPI assistant based on the identified design principles and evaluate the design in a controlled laboratory experiment. Within the laboratory experiment, we intend to evaluate the assistant's functionality (the distinct design principles in form of A/B tests), the end users satisfaction with the assistant (in form of a questionnaire) (Bhattacherjee 2001), and the assistant's perceived usability (in form of a questionnaire) (Albert and Tullis 2013). The experiment design will be adapted to the final UPI design, based on the derived design principles. Moreover, the experimental setting ensures a high internal validity and at the same time, as we use the test instance of CAMPSYS, ensures a reliable external validity (Bhattacherjee 2012). In a subsequent step, the CAMPSYS developers will be asked to assess one outcome of the experiment, the requirements articulated by the students, with respect to the requirements' quality (cf. Nidumolu 1996). We plan to test for the inter-rater reliability (Gwet 2014) by comparing the developers' conclusions for the articulated requirements. We will conclude the second design cycle with the analysis of the experiment's quantitative findings.



General Design Science Cycle	Design Cycle One	Design Cycle Two	Design Cycle Three	Design Cycle Four
Awareness of Problem	Literature review	Reflection of previous design cycle	Reflection of previous design cycle	Reflection of previous design cycle
Suggestion	Synthesis of design principles based on empirical findings	Refinement of design principles	Refinement of design principles	Refinement of design principles
Development	Instantiation of design principles as a prototype	Implementation of design principles as running protoype	Implementation of design principles as fully functional software application	Implementation of design principles as fully functional software application
Evaluation	Qualitative evaluation of prototype (focus groups)	Laboratory experiment	Longitudinal field study	Longitudinal field study
Conclusion	Focus group results analysis	Experiment results analysis	Field study results analysis Design theory	Field study results analysis Design theory (update)

Figure 1. Design science cycles (based on Kuechler and Vashniavi 2008)

The **third design cycle** aims for the evaluation of the UPI assistant in a longitudinal field study. Again, we will reflect the previous design cycles and refine the design according to the experiment's findings. Next, we will implement the assistant as a fully functional application and integrate it into the productive system of CAMPSYS. We aim to collect assistant usage data one the one hand, and the students' satisfaction (e.g. Bhattacherjee 2001) with the assistant and its usability (Albert and Tullis 2013) with an online-survey on the other hand. Moreover, we will continuously analyze the requirements' quality (cf. Nidumolu 1996) and developers' satisfaction (e.g. Bhattacherjee 2001) with the articulated issues (i.e. requirements and failure reports). Finally, we will analyze the findings from the field study and communicate them in form of a nascent design theory (Gregor and Hevner 2013; Jones and Gregor 2007) for an UPI assistant.

The **fourth design cycle** aims for the replication of the longitudinal field study in a company. We aim to cooperate with the software vendor of CAMPSYS and implement another UPI assistant instance for one of the vendor's SaaS solutions. Next, we plan to integrate the assistant into the SaaS solution and replicate the third design cycle's longitudinal field study. Finally, we will analyze the findings from the second field study and extent the design theory formulated in the third design cycle.

After describing our planned research activities in detail, we present the current state of our activities addressing the synthesis of meta-requirements and design principles for the end user involvement assistant in the following sections.

Towards Designing an End User Involvement Assistant

We aim to raise awareness for the potential benefits of involving end users in continued IS development. We identified three implications of previous research, whereas we address two of them in the current state of our research project. By involving the end users in continuous IS development, we expect a higher level of end user motivation (Wagner and Piccoli 2007). Nevertheless, subsequent to the current research project, there is the opportunity to investigate ways to motivate various types of users for an involvement in the continuous development of SaaS solutions. There are first indications for the success and usefulness of providing an assistant to foster continued end user participation (Yetim et al. 2012). What is missing is knowledge on the design of such UPI assistants and more empirical evaluations of their design as well as their outcomes. Thus, we target to gain deeper insights on the design of a user participation and involvement (UPI) assistant following a structured DSR approach ensuring both, high rigor and relevance of our research results (Hevner 2007). In the following, we discuss the current state of the derivation of our meta-requirements for a UPI assistant. We address the three implications identified in the conducted literature review. Furthermore, we ground the derived meta-requirements on assistance research and research on IS development.

First, for an ease of developers' understanding of the articulated requirements or failures, it is important to capture the current context of the user. In order to spare end users a lengthy description of what they are currently doing, which system functionality they are using, etc., we argue that the assistant should capture the current context of the users from a system perspective (Gregor and Benbasat 1999). Thus, we formulate our first meta-requirement (MR):

> **MR1**: Capture the current context of the users from a system perspective, e.g. which system functionality are they currently using, what is the related data etc.

When implementing MR1, the assistant will enable the end users to focus on the description of the requirement or failure. A very simple option to articulate requirements or failures is the form of a textual description. The assistant could provide some sort of interactive wikis combined with an auto completion feature supporting end users to articulate their needs with a common set of terms. However, textual descriptions have some limitations, such as limited expressiveness or the need to provide a lengthy description in order to describe a fact (Kuechler and Vaishnavi 2006). In order to support end users in articulating their needs, the assistant should provide supportive functionalities. There are various possibilities such as the integration of a screenshot functionality with an additional drawing functionality. Such a feature enables end users to support their textual description with some visual expressions. Another possibility would be the integration of a screen-recording functionality in order to record the end users interaction with the system and to show a failure prone or a desired behavior. In addition, the assistant should provide the functionality to record a spoken description of the end users' requirements (see Figure 2). In sum, we formulate the second meta-requirement:

> **MR2**: Assist users to express their requirements / failure report with computer-based support, such as an enterprise groupware system.

The identified third implication addresses the developers: *Enable developers to analyze a vast amount of complex end user feedback effectively.* Addressing this issue, we rely on existing concepts for the management of end user feedback and requests, an issue tracking system (cf. Serrano and Ciordia 2008) e.g. Jira. Issue tracking systems promise to categorize and manage, for example, requests in order to ease the handling of requirements and make them traceable. Therefore, we propose to combine the UPI assistant with the developers' issue tracking system in our third meta-requirement:

> **MR3**: Combination of an assistant with an issue tracking system to manage the requirements provided by the end users.

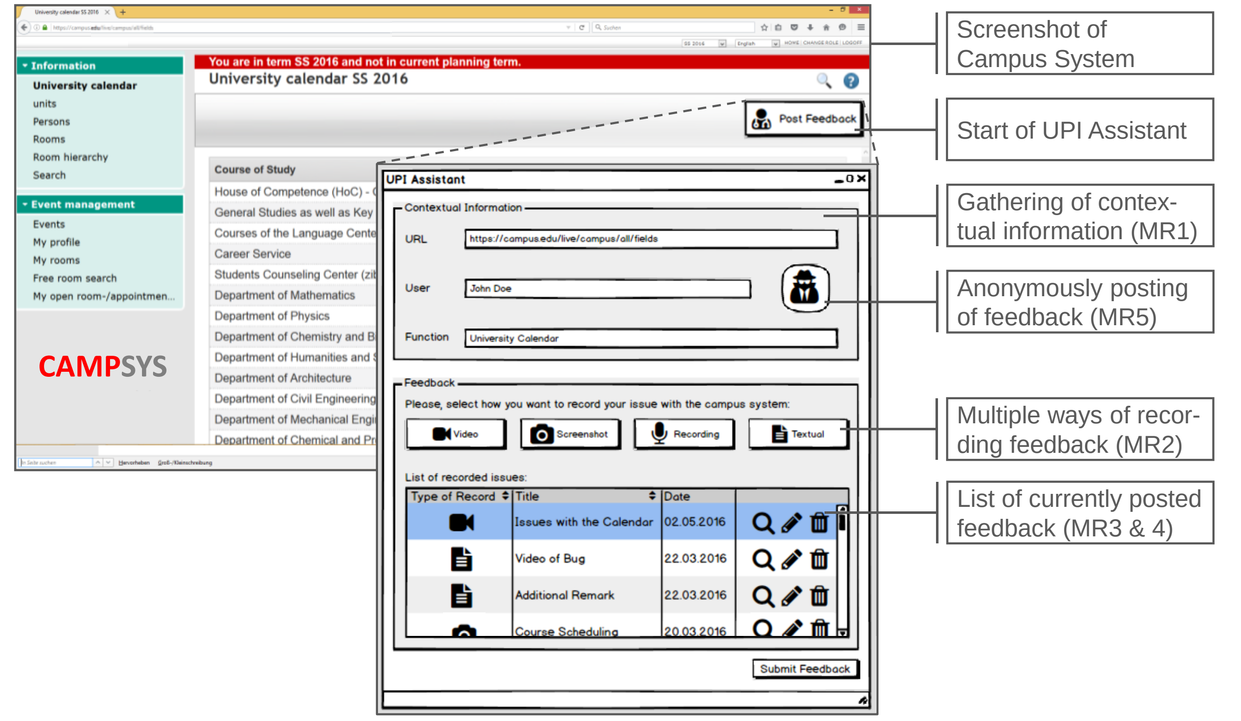
In addition, the assistant should automatically classify the created tickets. MR1 enables the assistant to be aware of the end user's current context. However, end users provide additional information while articulating their needs. For example, they convey whether the need is a failure report or a new requirement. A pre-defined, common set of terms can be applied to this information to classify the created ticket. This allows routing the ticket directly to the right team of developers.

> **MR4**: End user tickets are automatically classified according to the addressed system functions, type of ticket, and additional classification information.

Finally, the created ticket should contain information to contact the end users in case the developers require more information. Thus, the assistant should allow the communication between end users and developers. However, previous research has shown that it is crucial to consider information transparency in relation to users' willingness to be profiled (cf. Awad and Krishnan 2006). In order to secure private information, the assistant should enable the end users to decide, whether the ticket should contain their personal information for direct contact or whether they want to remain anonymous. This feature should enable a developer to contact the end user using the UPI assistant without knowing any personal information of the end user. In addition, end users can reject the communication feature if they do not want developers to contact them. Based on the aforementioned insights, we propose the fifth meta-requirement:

> **MR5**: Implement a developer-to-end user communication feature, but enable the end users to decide if the developers see their personal information. The assistant is supposed to enable anonymized communication or to reject communication outright.

Figure 2 gives a first impression of the UPI assistant highlighting the meta-requirements (MR) and its usage in form of a mock-up.


University calendar SS 2016
https://campus.edu/live/campus/all/fields
SS 2016
English
HOME | CHANGE ROLE | LOGOFF
Information
University calendar
units
Persons
Rooms
Room hierarchy
Search
Event management
Events
My profile
My rooms
Free room search
My open room-/appointmen...
CAMPSYS
Post Feedback
Course of Study
House of Competence (HoC) - (
General Studies as well as Key
Courses of the Language Cente
Career Service
Students Counseling Center (zi
Department of Mathematics
Department of Physics
Department of Chemistry and B
Department of Humanities and S
Department of Architecture
Department of Civil Engineering
Department of Mechanical Engi
Department of Chemical and Pr
Hervorheben
Groß-/Kleinschreibung
Video
Screenshot
Recording
Textual
Submit Feedback
Screenshot of Campus System
Start of UPI Assistant
Gathering of contextual information (MR1)
Anonymously posting of feedback (MR5)
Multiple ways of recording feedback (MR2)
List of currently posted feedback (MR3 & 4)

You are in term SS 2016 and not in current planning term.

University calendar SS 2016

UPI Assistant

Contextual Information

URL https://campus.edu/live/campus/all/fields

User John Doe

Function University Calendar

Feedback

Please, select how you want to record your issue with the campus system:

List of recorded issues:

Type of Record	Title	Date
🎥	Issues with the Calendar	02.05.2016
📄	Video of Bug	22.03.2016
📄	Additional Remark	22.03.2016
📷	Course Scheduling	20.03.2016

Figure 2. User participation and involvement assistant mock up

In our literature review, we identified three implications for user participation and involvement in continued IS development. The proposed design for an UPI assistant, in the form of five meta-requirements, aims to address these implications. The following Table 2 depicts how the proposed design maps to the implications. Please note, that the first implication from literature is not addressed specifically in the current stage of the research project. However, it can be argued that the instantiation of the first and second meta-requirement might also address the first implication.

Table 2. Implications and addressing meta-requirements

	Description	Meta-requirements				
		MR1	MR2	MR3	MR4	MR5
Implication 1	End users are more likely to be motivated as participants whenever they can provide feedback in their actual use context.	(✓)	(✓)			
Implication 2	End users need to be enabled to provide meaningful feedback to developers during use.	✓	✓		✓	
Implication 3	Developers need to be enabled to effectively analyze the vast amount of complex end user feedback.	✓		✓		✓

Expected Results and Outlook

This paper presents our ongoing DSR project on the end user involvement and participation in continued IS development. We conducted a systematic literature review to investigate UPI in continuous IS development and identified three implications for end user involvement from the literature. It is important to support UPI in the use context by enabling users to articulate meaningful issues and by enhancing developers' ability to make sense of it. Addressing this need, we outline our recently started DSR project aiming to design an UPI assistant. In the present paper, we raise problem awareness, outline the planned four design cycles, and discuss the current state of the meta-requirements derivation in design cycle one. From a research perspective, we contribute by addressing the theory-driven problem identification, that UPI for continuous development in SaaS is not addressed appropriately. We aim to develop a nascent design theory addressing the problem and to describe the design of an UPI assistant. Our research can be classified as an improvement according to the DSR contribution framework by Gregor and Hevner (2013). From a practical perspective, we will provide valuable design knowledge on how to design UPI assistants. Software vendors and organizations can implement their own UPI assistants for continued IS development context.

Although our research follows established guidelines for conducting literature reviews (Kitchenham and Charter 2007) and DSR projects (Kuechler and Vashnavi 2008), there are some limitations which need to be discussed. As with any literature review, there is a limit to the completeness of our sample of the literature. Furthermore, different system types could lead to different UPI contexts and therefore result in different characteristics of UPI (Cavaye 1995; Garrity 1994). The outlined DSR project is still in an early stage and we can only illustrate and discuss the intended design in form of meta-requirements and the proposed effects of the UPI assistant. Thus, the actual success of the assistant and its effects are subject for future research. Such results might help developers or project managers to derive strategies that are more effective for UPI in the continuous IS development context. Furthermore, we only partially address the first implication of our literature review that users will be more motivated, when they are able to provide feedback to developers in their actual use context. This is a limitation of our research, as it is possible that end users might not be motivated to use yet another tool. Instead, they might resist the use of a new design feature because they got 'too much other work to do'. Raising end user motivation to adopt a continuous IS development assistant can be the object of future research. Such research could, for instance, involve gamification mechanics and dynamics to improve the user experience of the UPI assistant.

Three studies in our literature review have pointed out the lack of prioritization control from end users before the implementation of a redesigned system, which influence the quality of an implementation (Begier 2010; Hansson et al. 2004, 2006). This is one particular aspect of common knowledge creation. Its quality is an important prerequisite for system success and depends on the effectiveness of the communication process between users and developers (Tesch et al. 2009). These aspects constitute important avenues for future research, which can be addressed with modified versions of our research approach. For instance, it would be possible to develop an assistant that cannot only guide users in their draft of tickets for the IT service organization, but also allows the prioritization of these requirements by end users.

References

Abelein, U., and Paech, B. 2013. “Understanding the Influence of User Participation and Involvement on System Success – a Systematic Mapping Study,” *Empirical Software Engineering* (20:1), pp. 28–81.

Albert, W., and Tullis, T. 2013. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*, Waltham, MA: Newnes.

Åsand, H.-R. H., and Mørch, A. I. 2006. “Super users and local developers: The organization of end-user development in an accounting company,” *Journal of Organizational and End User Computing* (18:4), pp. 1–21.

Awad, N. F., and Krishnan, M. S. 2006. “The Personalization Privacy Paradox : An Empirical Evaluation of Information Transparency,” *MIS Quarterly* (30:1), pp. 13–28.

Bano, M., and Zowghi, D. 2013. “User involvement in software development and system success,” in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering - EASE ’13*, New York, New York, USA: ACM Press, April 14, p. 125.

Barki, H., and Hartwick, J. 1989. “Rethinking the concept of user involvement,” *MIS quarterly* (13:1), pp. 53–63.

Barki, H., and Hartwick, J. 1994. “Measuring User Participation, User Involvement, and User Attitude.,” *MIS Quarterly* (18:1), pp. 59–82.

Begier, B. 2010. “Evolutionally improved quality of intelligent systems following their users’ point of view,” in *Advances in Intelligent Information and Database Systems*, Berlin Heidelberg: Springer, pp. 191–203.

Bellucci, A., Jacucci, G., Kotkavuor, V., Serim, B., Ahmed, I., and Ylirisku, S. 2015. “Extreme Co-design: Prototyping with and by the User for Appropriation of Web-connected Tags,” in *End-User Development,* P. Díaz, V. Pipek, C. Ardito, C. Jensen, I. Aedo, and A. Boden (eds.) (Vol. 9083), Cham, CH: Springer, pp. 109–124.

Bhattacherjee, A. 2001. “Understanding Information Systems Continuance: An Expectation-Confirmation Model,” *MIS Quarterly* (25:3), p. 351-370.

Bhattacherjee, A. 2012. *Social Science Research: Principles, Methods, and Practices Health research policy and systems / BioMed Central* (Vol. 9), Tampa, FL: USF Tampa Library Open Access Collections.

Bhattacherjee, A., and Lin, C.-P. 2014. “A unified model of IT continuance: three complementary perspectives and crossover effects,” *European Journal of Information Systems* (24:4), Nature Publishing Group, pp. 1–10.

Bhattacherjee, A., and Sanford, C. 2006. “Influence Processes for Information Technology Acceptance : An Elaboration Likelihood Model,” *MIS Quarterly* (30:4), pp. 805–825.

Bragge, J., and Merisalo-Rantanen, H. 2009. “Engineering E-Collaboration Processes to Obtain Innovative End-User Feedback on Advanced Web-Based Information Systems.,” *Journal of the Association for Information Systems*. (10:3), pp. 196–220.

Browne, G. J., and Rogich, M. B. 2001. “An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques,” *Journal of Management Information Systems* (17:4), pp. 223–249.

Carroll, J. 2004. “Completing Design in Use: Closing the Appropriation Cycle,” *European Conference of Information Systems, January 2004*, p. 11.

Cavaye, A. L. M. M. 1995. “User participation in system development revisited,” *Information & Management* (28:5), pp. 311–323.

Davenport, T. H. 1998. “Putting the Enterpirse into the Enterprise System,” *Harvard Business Review* (76:4), pp. 1–12.

DeSanctis, G., and Poole, M. S. 1994. “Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory,” *Organization Science* (5:2), pp. 121–147.

El Emam, K., Quintin, S., and Madhavji, N. H. 1996. “User Participation in the Requirements Engineering Process: An Empirical Study,” *Requirements Engineering* (1:1), pp. 4–26.

Finck, M., Gumm, D., and Pape, B. 2004. “Using Groupware for Mediated Feedback,” in *Proceedings of the Participatory Design Conference - PDC-04* (Vol. 2), pp. 45–48.

Fischer, G. 2008. “Rethinking software design in participation cultures,” *Automated Software Engineering* (15:3-4 SPEC. ISS.), pp. 365–377.

Foster, S. T. J., and Franz, C. R. 1999. “User involvement during information systems development: a comparison of analyst and user perceptions of system acceptance,” *Journal of Engineering and Technology Management* (16), pp. 329–348.

Garrity, E. J. 1994. “User Participation, Management Support and System Types,” *Information Resources Management Journal* (7:3), pp. 34–38.

Gregor, S., and Benbasat, I. 1999. “Explanations From Intelligent Systems: Theoretical Foundations and Implications for Practice”, *MIS Quarterly* (23:4), pp. 497–530.

Gregor, S., and Hevner, A. R. 2013. “Positioning and presenting design science research for maximum impact,” *MIS Quarterly* (37:2), pp. 337–355.

Gwet, K. L. 2014. *Handbook of Inter-Rater Reliability: The Definitive Guide to Measuring The Extent of Agreement Among Raters* (4th ed.), Gaithersburg, MD.

Hansson, C., Dittrich, Y., and Randall, D. 2004. “Agile Processes Enhancing User Participation for Small Providers of Off-the-Shelf Software,” in *Extreme Programming and Agile Processes in Software Engineering: Proceedings of the 5th International Conference, XP 2004*, pp. 175–183.

Hansson, C., Dittrich, Y., and Randall, D. 2006. “How to include users in the development of off-the-shelf software: A case for complementing participatory design with agile development,” in *Proceedings of the Annual Hawaii International Conference on System Sciences* (Vol. 8), pp. 1–10.

Harris, M., and Weistroffer, H. 2009. *A New Look at the Relationship between User Involvement in Systems Development and System Success Communications of the Association for Information Systems* (Vol. 24), pp. 739–756.

Hendry, D. G. 2008. “Public participation in proprietary software development through user roles and discourse,” *International Journal of Human Computer Studies* (66:7), pp. 545–557.

Hevner, A. R. 2007. “A Three Cycle View of Design Science Research,” *Scandinavian Journal of Information Systems* (19:2), pp. 87–92.

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

Hsieh, J. P.-A., and Wang, W. 2007. “Explaining employees’ Extended Use of complex information systems,” *European Journal of Information Systems* (16:3), pp. 216–227.

Hwang, M., and Thorn, R. 1999. “The effect of user engagement on system success: a meta-analytical integration of research findings,” *Information & Management* (35:4), p. 230.

Ives, B., and Olson, M. H. 1984. “User Involvement and MIS Success: A Review of Research,” *Management Science. May84* (30:5), pp. 586 – 603.

Jasperson, J. (Sean), Carter, P. E., and Zmud, R. W. 2005. “A Comprehensive Conceptualization of Post-Adoptive Behaviors Associated with Information Technology Enabled Work Systems,” *MIS Quarterly* (29:3), pp. 525–557.

Jones, D., and Gregor, S. 2007. “The anatomy of a design theory,” *Journal of the Association for Information Systems* (8:5), p. 312-335.

Just, S., Premraj, R., and Zimmermann, T. 2008. “Towards the next generation of bug tracking systems,” in *Proceedings - 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2008*, pp. 82–85.

Kabbedijk, J., Brinkkemper, S., Jansen, S., and Van Der Veldt, B. 2009. “Customer involvement in requirements management: Lessons from mass market software development,” in *Proceedings of the IEEE International Conference on Requirements Engineering*, pp. 281–286.

Kappelman, L. A. 1995. “Measuring user involvement: a diffusion of innovation perspective,” *Database Advances* (26:2), pp. 65–86.

Karahanna, E., Straub, D. W., and Chervany, N. L. 1999. “Information Technology Adoption Across Time: A Cross-Sectional Comparison of Pre-Adoption and Post-Adoption Beliefs,” *MIS Quarterly*(23:2), pp. 183–213.

Kawalek, P., and Wood-Harper, T. 2002. “The Finding of Thorns: User Participation in Enterprise System Implementation,” *ACM SIGMIS Database - Special issue on Critical analysis of ERP systems: the macro level* (33:1), pp. 13–22.

Kim, Y., and Crowston, K. 2011. “Technology Adoption and Use Theory Review for Studying Scientists ’ Continued Use of Cyber-infrastructure,” *Proceedings of the American Society for Information Science and Technology* (48:1), pp. 1–10.

Kitchenham, B., and Charters, S. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*.

Kuechler, B., and Vaishnavi, V. K. 2008. “On theory development in design science research: anatomy of a research project.,” *European Journal of Information Systems* (17:5), pp. 489–504.

Kuechler, W. L., and Vaishnavi, V. 2006. “So, talk to me: The effect of explicit goals on the comprehension of business process narratives.,” *MIS Quarterly* (30:4), pp. 961–979.

Kujala, S. 2008. “Effective user involvement in product development by improving the analysis of user needs,” *Behaviour & Information Technology* (27:6), pp. 457–473.

Lieberman, H., Paternò, F., Klann, M., and Wulf, V. 2006. *End-user development: An emerging paradigm End User Development SE - 1* (Vol. 9), Springer Netherlands.

Lu, H.-P., and Wang, J.-Y. 1997. “The relationships between management styles, user participation, and system success over MIS growth stages,” *Information & Management* (32:4), pp. 203–213.

Mabert, V. A., Soni, A., and Venkataramanan, M. A. 2001. “Enterprise resource planning: Common myths versus evolving reality,” *Business Horizons* (44:3), pp. 69–76.

Maedche, A., Morana, S., Schacht, S., Werth, D., and Krumeich, J. 2016. "Advanced User Assistance Systems," *Business & Information Systems Engineering, pp. 1-4.*

Majchrzak, A., Rice, R. E., Malhotra, A., King, N., and Ba, S. 2000. “Technology Adaptation: The Case of a Computer-Supported Inter-Organizational Virtual Team,” *MIS Quarterly* (24:4), pp. 569–600.

Nidumolu, S. 1996. “A comparison of the structural contingency and risk-based perspectives on coordination in software-development projects,” *Journal of Management Information Systems* (13:2), pp. 77–113.

Olson, M. H., and Ives, B. 1981. “User Involvement in System Design: An Empirical Test of Alternative Approaches,” *Information & Management* (4:4), pp. 183–195.

Orlikowski, W., and D. Hoffman. 1997. "An Imporvisational Model for Change Managment: The Case of Groupware Technologies." in *Inventing the Organizations of the 21st Century,* T.W. Malone, R. Laubacher, M.S. Scott Morton (eds.), Boston, MA: MIT Press , pp. 265–282.

Orlikowski, W. J. 2000. “Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations,” *Organization Science* (11:4), pp. 404–428.

Pagano, D., and Bruegge, B. 2013. “User involvement in software evolution practice: A case study,” in *2013 35th International Conference on Software Engineering (ICSE)*, pp. 953–962.

Pagano, D., and Maalej, W. 2013. “User feedback in the appstore: An empirical study,” in *2013 21st IEEE International Requirements Engineering Conference, RE 2013 - Proceedings*, pp. 125–134.

Parmentier, G., and Gandia, R. 2013. “Managing Sustainable Innovation with a User Community Toolkit: The Case of the Video Game Trackmania,” *Creativity and Innovation Management* (22:2), pp. 195–208.

Qin, M., and Xu, S. 2007. “Understanding information technology preadoption and postadoption: An integrated process model,” in *2007 International Conference on Management Science and Engineering*, pp. 239–244.

Saeed, K. A., and Abdinnour, S. 2013. “Understanding post-adoption IS usage stages: an empirical assessment of self-service information systems,” *Information Systems Journal* (23:3), pp. 219–244.

Serrano, N., and Ciordia, I. 2008. “Bugzilla, ITracker, and Other Bug Trackers,” *IEEE Software* (22:2), pp. 11–13.

Tesch, D., Sobol, M. G., Klein, G., and Jiang, J. J. 2009. “User and developer common knowledge: Effect on the success of information system development projects,” *International Journal of Project Management* (27:7), pp. 657–664.

Tremblay, M. C., Hevner, A. R., and Berndt, D. J. 2010. “The Use of Focus Groups in Design Science Research,” in *Design Research in information Systems*A. R. Hevner and S. Chatterjee (eds.) (Vol. 22), Boston, MA: Springer US, pp. 121–143.

Wandke, H. 2005. "Assistance in human–machine interaction: A conceptual framework and a proposal for a taxonomy", *Theoretical Issues in Ergonomics Science*, (6:2), pp. 129–155.

Wagner, E. L., and Piccoli, G. 2007. “Moving beyond user participation to achieve successful IS design,” *Communications of the ACM* (50:12), pp. 51–56.

Wilson, R., Maniatopoulos, G., Martin, M., and McLoughlin, I. 2012. “Innovating Relationships: taking a co-productive approach to the shaping of telecare services for older people,” *Information, Communication & Society* (15:7), pp. 1136–1163.

Wixom, B. H., and Todd, P. A. 2005. “A theoretical integration of user satisfaction and technology acceptance,” *Information Systems Research* (16:1), pp. 85–102.

Yang, X., and Plewe, D. A. (2016). "Assistance Systems in Manufacturing: A Systematic Review". in *Advances in Intelligent Systems and Computing. Advances in Ergonomics of Manufacturing: Managing the Enterprise of the Future,* Schlick, C. and S. Trzcieliński (eds.), Cham, CH: Springer International Publishing pp. 279–289.

Yetim, F., Draxler, S., Stevens, G., and Wulf, V. 2012. “Forstering Continuous User Participation by Embedding A Communication Support Tool in User Interfaces,” *Transactions on Human-Computer Interaction* (4:2), pp. 153–168.

Yetton, P., martin, A., Sharma, R., and Johnston, K. 2000. “A model of information systems development project performance,” *Information Systems Journal* (10:4), pp. 263–289.

Zhou, J., Zhang, H., and Lo, D. 2012. “Where Should the Bugs Be Fixed?,” in *Proceedings of the 34th International Conference on Software Engineering*, pp. 14–24.