HOW TO DESIGN AN INTERNAL CROWDSOURCING SYSTEM

Short Paper

Nicolas Knop
University of St. Gallen
Institute of Information Management
Müller-Friedberg-Strasse 8, CH-9000
St. Gallen, Switzerland
nicolas.knop@unisg.ch

David Durward
University of Kassel
Research Center for ISDesign (ITeG)
Pfannkuchen-Strasse 1, D-34121
Kassel, Germany
david.durward@uni-kassel.de

Ivo Blohm
University of St. Gallen
Institute of Information Management
Müller-Friedberg-Strasse 8, CH-9000 St. Gallen, Switzerland
ivo.blohm@unisg.ch

Abstract

Digitalization gives rise to dynamic forces shaping future working structures. In practice, companies are increasingly interested in using their own employees as an internal crowd. Drawing on socio-technical systems (STS) perspective, organizations must understand how they can embed an internal crowdsourcing system effectively in order to exploit its potential. In particular, organizations need assistance in designing internal crowdsourcing systems. Thus, we follow an action design research approach and develop comprehensive design guidelines for designing an internal crowdsourcing system. The paper in hand develops the design guidelines by deriving the requirements from literature, developing them further in a bank project and evaluating them with external experts. Furthermore, we assess what tools can cover which part of the system and implement the system. The short paper will present the derived requirements, the first design of the internal crowdsourcing system and explain next steps of the full paper as well as the contribution.

Keywords: Crowdsourcing, Design Guidelines, Software Testing, Social-Technical Systems

Introduction

The use of crowdsourcing has increased substantially in recent years since it describes a promising alternative to traditional employment in today's digital era (Kuek et al. 2015). It has emerged as a new form of orchestrating work that is based on certain principles. According to Blohm et al. (2013), the fundamental idea of crowdsourcing is that a crowdsourcer (which could be a company, an institution or a non-profit organization) proposes to an undefined group of contributors or crowd workers (individuals, formal or informal teams, other companies) the voluntary undertaking of a task presented in an open call. For companies, it provides several benefits such as broader access to specialized skills (Prpić et al. 2015), more flexible and faster hiring processes (Kuek et al. 2015), shorter product development cycles (Simula 2013) as well as relatively low costs (Schenk et al. 2011). However, crowdsourcing can be applied within and beyond the organizational boundaries of a crowdsourcer. In the external setting, crowd workers include individuals from outside the company boundaries that participate in some sort of digital freelancing. Nevertheless, due to the numerous advantages, companies increasingly use their own employees as an internal crowd to leverage the collective intelligence and further orchestrate them more efficiently for certain tasks. In practice, this concept of *internal crowdsourcing* is nowadays used for various activities within business processes e.g., (Bailey et al. 2010; Rohrbeck et al. 2015).

Since the phenomenon of internal crowdsourcing involve complex interactions between humans, technology and environmental factors (Baxter et al. 2011), it describes novel socio-technical systems (STS). Within internal crowdsourcing, social and technical factors create conditions that determine organizational performance. By drawing on STS theory (Leavitt 1964; Lyytinen et al. 2008), we are able to examine internal crowdsourcing as systems possessing five interacting components: task; structure; actor; technology and environment (Beese et al. 2015; Weilbach et al. 2015). Furthermore, research have explored STS issues that affect the specification, design and operation of work (Baxter et al. 2011). However, in order to implement internal crowdsourcing successfully within the organization, research findings on external forms cannot be directly applied since there are structural differences. Regarding internal crowdsourcing, the working relationships between crowdsourcer and crowd worker are determined by employment contracts (Simula et al. 2012). Therefore, in contrast to external crowd workers, the employees perform crowdsourced tasks within the scope of their main activities and are not able to choose their own working time and location freely (Bonabeau 2009; Lopez et al. 2010). Despite the growing interest, internal crowdsourcing still bears some important research gaps. First research has shown why organizations should apply internal crowdsourcing e.g. (Ågerfalk et al. 2008). Furthermore, early studies focused on the benefits of using internal crowdsourcing, such as the fast access to internal knowledge (Gaspoz 2011), the co-creation of enterprise strategy (Jette et al. 2015) or idea generation and selection (Soukhoroukova et al. 2012). Although the advantages have been identified, companies still do not know how they can embed an effective crowdsourcing process internally (Fitzgerald et al. 2015) in order to exploit its benefits (Zuchowski et al. 2016). The socio-technical components and its interrelations have been largely neglected in previous crowdsourcing research. In particular, the understanding of how to design an internal crowdsourcing system is still a black box in recent studies. Thus, we provide comprehensive design guidelines in order to enable companies to benefit from its internal crowdsourcing systems. Therefore, we intend to fill the outlined research gaps by addressing the following research question: *How can an effective internal crowdsourcing system be designed?*

In this paper, we address these issues by drawing on an action design research project in which we design an internal crowdsourcing system together with a Swiss bank. We therefore draw on the well-established framework of STS theory and its components. Our results will be formalized in specific design guidelines in order to support researchers and practitioners in designing effective internal crowdsourcing systems. We conduct our research in the context of software testing since this case offers many benefits. First, testing can illustrate greatly the benefit of a crowd, using internal knowledge effectively. For instance, given enough eyeballs, all bugs are shallow. Second, software testing describes a labor-intensive regular task, giving sufficient opportunities to observe and develop the crowdsourcing system. Moreover, while software quality is gaining in importance the traditional testing approaches reach their performance limits. Companies face an increasingly dynamic environment with shorter product lifecycles and higher diversity of hardware devices, increasing the complexity of software development, which leaves traditional testing less applicable (Leicht et al. 2017; Leicht et al. 2016a). Hence, our findings are particularly relevant in crowdsourcing settings that include testing activities and processes, e.g., *software testing.*

This paper proceeds as follow: Section two provides the theoretical background by introducing related work. In section three, we outline the applied methodology. Afterwards, we present our preliminary results of the study. Finally, we illustrate the expected contributions after presenting our next steps.

Related Work

Socio-Technical Systems Theory

The underlying premise of STS theory describes the fact that systems design should be a process that takes into account both social and technical factors influencing the functionality and usage of IT-based systems (Baxter et al. 2011). The overall behavior of such kind of systems depends on a diverse set of often non-linear and dynamic mechanisms that relate to both the social and technical subsystems (Beese et al. 2015). The theoretical lens of STS is probably the most extensive body of conceptual and empirical literature underlying work design applications today (e.g., Mumford 2006; Sykes et al. 2014). In literature, researchers applied STS theory to analyze systems that involve a complex interaction between humans, technology and the environmental aspects of the work system (Baxter et al. 2011). From STS perspective, these work systems consist of five basic socio-technical components (i.e., *tasks*, *actors*, *structure*, *technology*, and *environment)* (Beese et al. 2015; Lyytinen et al. 2008). According to Lyytinen and Newman (2008), *tasks*, *actors*, *structure*, and *technology* interact with each other and are embedded in the organizational *environment* that is driving and influencing change (Beese et al. 2015).

However, following recent studies, information systems (IS) can be seen as complex STS in which humans, being part of social subsystems, and IT artifacts, being part of technical subsystems, interact to process information (Beese et al. 2015; Lyytinen et al. 2008). In particular, crowdsourcing describes a special case of IS that produce informational products and/or services for internal or external customers by harnessing the potential of crowds (Geiger et al. 2014). Against this backdrop, we believe the STS perspective to be an appropriate theoretical lens to examine internal crowdsourcing systems in order to derive design guidelines for this novel form of work organization.

Internal Crowdsourcing as a Socio-Technical System

According to (Zuchowski et al. 2016), internal crowdsourcing is an IT-enabled group activity based on an open call for participation in an enterprise. Based on this definition and the notion of STS theory, internal crowdsourcing describes a complex work system. First, it represents a collaborative, competitive, or networked group-activity of various *humans* (Zhu et al. 2014). Second, it is a *technology*-enabled phenomenon using both generic social media (e.g., wikis, blogs) (Stocker et al. 2012) and dedicated specialist tools (e.g., a scanning tool for weak signals on change to support the corporate foresight activities) (Rohrbeck et al. 2015). Third, internal crowdsourcing takes place in organizational settings as a specific *environment* (Simula et al. 2012). Thus, based on prior IS research (Beese et al. 2015; Lyytinen et al. 2008), we outline the five components of internal crowdsourcing as a STS:

- *Actors:* Includes organization's members, its main stakeholders who carry out the internal crowdsourcing projects, their influences and motivation.
- *Task:* Describes the internal crowdsourcing systems goals and purpose and the way in which the crowdsourcing gets done within the organization.
- *Structure:* The structure covers systems of communication, systems of authority, and systems of workflow. It further includes both the normative dimension, that is, values, norms, and general role expectations, and the behavioral dimension, that is, the patterns of behavior as actors communicate, exercise authority, or work within the internal crowd.
- *Technology*: Technology denotes tools – problem-solving inventions like work measurement and computers that compose part of the internal crowdsourcing system.
- *Environment:* Describes to what extent the internal crowdsourcing takes place in a certain organizational setting.

However, there are organizational circumstances that act as potential barriers to acceptance and use of internal crowdsourcing systems and thereby affect the five components mentioned above (Erickson et al. 2012). For example, the protection of hierarchical status and resistance from certain departments may influence the structure of an internal crowdsourcing system. With regard to the actors of the internal crowdsourcing system, there might exist entry barriers that limit the potential amount of contributors.

Furthermore, the specific context of internal crowdsourcing needs to be examined in order to design the above mentioned STS components and determine which parameters are the most suitable for particular situations. Software testing is a major application field in practice since the rapid development of new IT-enabled business models and a fast growing hardware market (Leicht et al. 2016b; Zogaj et al. 2014b). Thus, we apply the dynamic STS perspective and examine the components of internal crowdsourcing as well as its interrelations in the specific context of software testing.

Method

Action Design Research

In order to develop the design guidelines for internal crowdsourcing systems, we apply action design research (ADR) (Sein et al. 2011). This research method develops solutions for classes of problems that contain relevance for practice and allows systematic specification of design knowledge, based on a real life problem (Gregor et al. 2007; Von Alan et al. 2004). It strives for a common understanding by collaborating directly and closely with practitioners in order to improve iteratively IT artifacts that solve the given problem (Sein et al. 2011). It combines theoretical understanding of research with practical and situated knowledge in order to deepen the understanding and study the real life problem in its ecosystem (Davison 2001; Kohler et al. 2011; Street et al. 2004). ADR generates prescriptive knowledge through developing, evaluating, and reflecting IT artifacts within organizational research contexts. Finally, it develops the generated knowledge further, which allows deriving design guidelines for the investigated problem class (Giessmann et al. 2016; Sein et al. 2011). Consequently, this paper relies on Sein et al.'s (2011) conceptualization of the IT artifact. ADR is the most suitable method to achieve our goals, since we want to address a class of problems (Bitzer et al. 2016). In our case, developing design guidelines for conceptualizing internal crowdsourcing systems for software testing in a Swiss bank. As ADR allows systematic specification of design knowledge (Gregor et al. 2007; Von Alan et al. 2004), we believe that we can derive robust design guidelines for internal crowdsourcing systems. In addition, we suggest that the – complementary and supporting – combination of theory and practice will enable us to address our goals appropriately due to comprehensive knowledge.

Project setting

The ADR project is pursued with a Swiss bank that is currently developing new software that will be used by every employee for daily business operations. The company has a long tradition in integrating its end users (i.e., employees in the single banks working with the system) in the testing of its enterprise applications that the bank used to invite to the headquarter for testing. Nevertheless, the bank faces the problem that the current testing process does not integrate the employees efficiently enough to reach an adequate test coverage and software quality, as explained further down in the section problem formulation. Consequently, the ADR project intends to increase effectiveness and efficiency of end user testing by introducing the principle of crowdsourcing and build an internal crowdsourcing system for software testing. It will integrate employees into a testing process efficiently; provide access to tests, a platform for the documentation and evaluation. The Swiss bank did not possess design knowledge in regards to internal crowdsourcing systems. For this reason, the bank set up an interdisciplinary team that consists of crowd testing researchers, operative test managers, higher test management, and defect management experts, as well as test service delivery specialists in September 2016. In order to support the bank's conceptualization of the internal crowdsourcing system and formalize the learning in design guidelines, we structured the research project according to the ADR method, as described in the next section. First, we defined the problem of the Swiss bank of integrating up to 215 internal crowd workers efficiently into the test process in the *Problem Formulation* phase and derived the goal state of the crowdsourcing system. Second, we will conduct in the *Building, Intervention and Evaluation (BIE)* phase three BIE cycles in order to derive requirements, conceptualize and design the system. After the last BIE cycle, we will summarize the findings and assess how the system addresses the problems of the bank in the *Reflection and Learning* phase. Finally, we plan to derive design guidelines in the *Specified Learning* phase, by abstracting the findings summarized in the *Reflection and Learning phase*, adding to the body of literature on how to design an internal crowdsourcing system and practitioners to use its potential.

Preliminary Findings

Problem Formulation

Our research is driven by the Swiss banks' need of a new and effective testing method as an addition to the conventional testing. In order to develop an understanding of the organizational problem and defining the challenges, we conducted two interviews with two of the banks' senior managers, responsible for the crowd testing project and a workshop of two hours with three crowdsourcing researchers as well as five testing experts. First, the bank had not enough resources for an adequate coverage of diverse fields, leaving an increased risk of potential mistakes in the company's software. The department did not have the resources to conduct sufficient regression tests after the launch of a product. The experts planned to address this problem, by expanding the coverage of the testing activities through increasing the amount of testers. The crowdsourcing system would include a crowd of 215 employees efficiently, eliminating potential blind spots regarding the software's functionality. Second, the test department needed more and faster testing cycles. So far, the testing department was not able to set testing cycles shortly and quickly. Some test cycles took too long to prepare or due to the lack of resources test cycles had to be postponed decreasing the quality. In order to address this, the bank intended to increase the speed of execution for single testing cycles for getting feedback concerning the quality of the software on a weekly basis through crowd testing. Third, the quality of the software needed to be increased by integrating end users and to extend the use of internal know how. At the current state, important software issues are detected at a late stage of the testing process, due to the end users get integrated late in testing. The bank could not integrate employees efficiently from different branches located all over the country. In the present situation employees have to travel hours to the headquarter in order to participate in internal testing. The bank addresses this, by involving employees coming from different areas of Switzerland through the internet-based nature of crowdsourcing. Employees would be able to join the testing process via their computers at their normal working space. The challenges of the Swiss bank are depicted in table 1.

Table 1. Challenges of the Swiss bank

No.	Challenge	Description
1	Adequate coverage	Insufficient coverage of software testing due to lack of resources and interdisciplinary knowledge of the conventional testers
2	Faster test cycles	Inability of preparing and conducting test cycles quickly, as well as setting a suitable amount of test cycles
3	Efficient integration of employees	Integrating employees efficiently to the test process, in order to use expert knowledge identifying software issues early

The paper addresses a class of problems, which is the real life problem of our Swiss bank, designing an internal crowdsourcing system for software testing to face its challenges in an increasingly dynamic environment and higher complexity of software development effectively. However, companies struggle to integrate the broad input of employees efficiently to gain appropriate coverage in fast product cycles. Designing an effective crowdsourcing system is difficult, concrete insights about what components are necessary and how to design them are missing.

Building, intervention, and evaluation

In the BIE phase, we will conduct three cycles. In the first BIE cycle, we derived the requirements from literature and practice for designing an internal crowdsourcing system. In the second BIE cycle, we focused on defining, deriving and designing the crowdsourcing system by combining the single components evaluated by test managers. In the third BIE cycle, we plan to implement the improvements derived from the evaluation in the previous cycle and evaluate the improved system with test managers, crowd testers and data from several test iterations, in order to capture the notion of effectivity.

First BIE cycle - Requirements

In the first BIE cycle, we derived requirements for a crowdsourcing system by conducting a literature review and a series of workshops with the Swiss bank. We validated the set of requirements by interviewing external Crowd testing experts.

The literature review revealed specific requirements for crowdsourcing systems, which we structured according to the STS components. This way the paper addressed the important basic components of a crowdsourcing system. We conducted the literature review according to (Webster et al. 2002) and (Vom Brocke et al. 2009). We searched in six databases (Business Source Premier, Econ Lit, JStor, Science Direct database, AIS electronic library and the ACM digital library). We used keywords "Crowdsourcing" and "System" or "Crowd Work System" or "Crowd testing". Papers were included in our analysis when they addressed one or more of the basic components of STS. In total, we reviewed 40 papers and 16 of them discussed requirements of crowdsourcing systems. As a result, we derived ten specific requirements.

The ten requirements found in the literature review, were refined and extended in two workshops at the Swiss bank (each around 120 minutes). Participants of these workshops were three researchers with vast experience in crowd testing, a test management expert, a defect management expert, and two test service delivery specialists. The group of experts assessed the 10 requirements found in the literature review, further detailed them and added five new requirements (italic writing in figure 1), leading to 15 requirements in total. Finally, we validated the obtained requirements with three external crowd testing experts who use crowd testing in their own organizations on a monthly basis. They evaluated our list of requirements in 30 to 45 minutes interviews. The experts agreed on our 15 requirements but added details to the existing ones. The requirements are depicted in figure 1, which lead to designing the building blocks of the associated STS components.

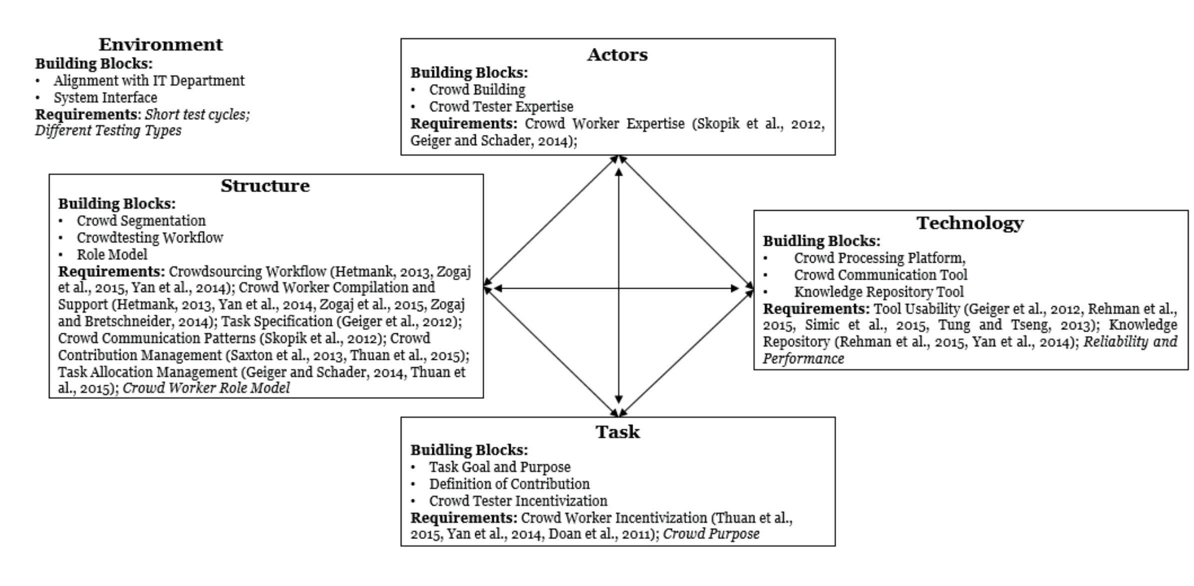

Figure 1. Internal Crowdsourcing System

Second BIE cycle – Internal Crowdsourcing System Design

In the second BIE cycle, we conceptualized the building blocks of the internal crowdsourcing system, based on the requirements derived in the first BIE cycle. A team of service deliverable experts, managers of the testing department and crowdsourcing researchers conducted four internal workshops (each around 120 minutes) in order to conceptualize the system's single building blocks and their interactions. They derived the requirements for the system and allocated each to a specific STS component according to the best fit of a STS component definition. Afterwards, they derived building blocks of the system based on a specific requirement by further detailing the requirements to the level of a first concept building block. Therefore, the building blocks share the STS component with the requirement they descent from. For instance, based on the requirement "Crowdsourcing Workflow", they further detailed and added steps to the workflow comprehensively as well as derived a role model accordingly as building blocks. Afterwards they assessed how other blocks interact. For example, the building block "Crowd Tester Expertise" of the "Actor" component interacts also with the "Crowd Testing Workflow" of the "Structure" component. The more suitable the crowd's expertise for a task, the more efficient the crowd can pass through the workflow solving the task." We illustrate the STS components, their building blocks and the requirements they base on in figure 1. In table 2 below, we describe the five STS components and their building blocks in detail.

Table 2. Building Blocks of Internal Crowdsourcing System

STS Component	Building Block	Description
Structure	Crowd Segmentation	Forming employee groups who signed up according to predefined criteria, in order to allocate an appropriate group of testers to a task.
	Crowd Testing Workflow	Designing an overarching workflow that describes the process on how single crowd tests are performed within the system. The process consists of 5 phases: **1.** Crowd workers receive invitations for tests, which they can accept or not. **2.** Tasks get allocated to the crowd by matching its criteria. **3.** Crowd tests the bank's software and documents all defects with support if needed. **4.** Input gets evaluated. **5.** Input gets filtered and the relevant input imported to the conventional IT department.
	Role Model	Defining different roles and responsibilities for the different actors in the crowdsourcing system as well as portioning decision rights. The role model consists of the following roles. *PMO (Project Management Officer):* Responsible for preparation and tests, includes task specification, selecting group of testers for test. *Test Manager:* Executes test initiatives. *Crowd Tester:* Conducts the testing. *Defect Manager:* Evaluates the results of the crowd.
Actors	Crowd Building	An announcement on the intranet platform of the company, asking to join the crowd. After the bank built up a crowd of 215 employees, it took down the announcement due to sufficient participants.
	Crowd Tester Expertise	The crowd testers have diverse expertise due to using different parts of the software in their daily business. A crowd tester is selected for a test according to his expertise in order to conduct the test effectively.
Technology	Crowd Processing Platform	Grants access to the test cases of the test iterations and enables the work processing. It provides possibilities for documenting defects as well as evaluating the input of the crowd.
	Crowd Communication tools	Communication tools send invitations for certain tests to predefined receivers and provide an overview of who accepted the invitation or not. In addition, e-mails, chat messaging and video calls are possible.

	Knowledge Repository	The crowd can access an area on the intranet, where different information is available to the crowd. For instance, information concerning dates of test iterations, contact information, summaries of results, templates or explanations of tools and the test process.
Task	Task Goal and Purpose	The purpose of the task is to receive input from the crowd regarding defects of the software based on their practical experience, missing in the IT department.
	Definition of Contribution	Internal crowdsourcing systems can produce different contributions. One has to define the intended contribution and design the system accordingly in order to increase effectivity of the system. We define the type of our task contribution; first, as heterogeneous, divers contributions of the crowd are possible. A tester finds unique defects, which are not found by other testers. Second, contributions are independent, a defect found has a value on its own; the value of a single defect does not arise only through combining all defects.
	Crowd Tester Incentivization	The crowd is mainly incentivized by the incentive mechanism “goal alignment”, between the personal goal of the employees and the project goal. After the testing and implementation phase, the employees will have to use the software in daily business, which results in two personal goals that are aligned with the project goal. First, the employees wish to gain early experience, in order to be prepared for the implementation of the software and to facilitate the transition from old to new software. Second, by being part of the test process, the employees have the opportunity to give input to increase the usability of the software and improve the software they will have to use in the future. Therefore, the incentive mechanism addresses an intrinsic motivation. In contrast, no financial incentive is given to the crowd.
Environment	Alignment with IT Department	Internal crowd test iterations are aligned with the test iterations of the IT department, because the crowd tests are used as a complementing part in the general testing efforts. For instance, when practical expertise of the crowd is needed for a test iteration, the IT department requests crowd tests.
	System Interface	Automated interface transfers test cases or defect reports of the crowd when needed, between the crowd test system and the IT department. For instance, for the crowd test preparation the test cases get transferred from the IT department to the crowd processing platform.

Next Steps and expected Contributions

First, we will evaluate the *second BIE*, by conducting an internal pre-test with six test managers that were not involved in designing the system. In addition, we will conduct workshops evaluating the system with testing experts and crowdsourcing researchers. Second, we will address the *final BIE cycle* by implementing the results of the previous evaluation and design the system. Afterwards, we will conduct 12 crowd test iterations, continuously evaluate and develop the system further. For this purpose, we will conduct semi-structured interviews with PMO’s, test and defect managers, a survey with crowd testers as well as evaluate data from the iterations. Furthermore, we will summarize the findings, assess how the system addresses the problems of the bank and captures the notion of effectivity of the system in the *Reflection and Learning* phase. Finally, we will derive design guidelines in the *Specified Learning* phase. We will abstract the findings summarized from the *Reflection and Learning phase* and add to the body of literature on how to design an internal crowdsourcing system as well as practitioners to use its potential.

Companies must understand how they can design and embed a crowdsourcing system internally (Fitzgerald et al. 2015), in order to use the potential of internal crowdsourcing effectively, such as the fast access to internal knowledge (Gaspoz 2011), use of co-creation (Jette et al. 2015) or idea generation and selection (Soukhoroukova et al. 2012). To our knowledge, this study is the first to address this gap by summarizing and deriving requirements for an internal crowdsourcing system comprehensively, designing an effective version of such a system and offering design guidelines, which are particularly relevant in crowdsourcing settings that include testing activities and processes i.e., software testing. The main contributions of this short paper reflect on one hand the comprehensive set of requirements, derived in the first BIE cycle. They are a basis for comprehensive design guidelines. On the other hand, it is presented by the initial concept of an internal crowdsourcing system for software testing and its building blocks from the second BIE cycle. The concept of the system combines insights from the available literature (Hetmank 2013; Skopik et al. 2012; Zogaj et al. 2015), the engaged project team and external crowdsourcing experts. This combination expand existing research regarding how to design internal crowdsourcing systems (Alter 2013; Geiger et al. 2012). Furthermore, our full paper will provide an effective internal crowdsourcing system design, comprehensively evaluated in 12 test iterations in the third BIE cycle. Finally, we will derive design guidelines that have the capability of explaining how such a system is designed effectively. They will enable practitioners to design an internal crowdsourcing system that exploits the benefits of internal crowdsourcing.

References

Ågerfalk, P.J., and Fitzgerald, B. "Outsourcing to an Unknown Workforce: Exploring Opensurcing as a Global Sourcing Strategy," *Mis Quarterly*) 2008, pp 385-409.

Alter, S. "Work system theory: overview of core concepts, extensions, and challenges for the future," *Journal of the Association for Information Systems* (14:2) 2013, p 72.

Bailey, B.P., and Horvitz, E. "What's your idea?: a case study of a grassroots innovation pipeline within a large software company," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2010, pp. 2065-2074.

Baxter, G., and Sommerville, I. "Socio-technical systems: From design methods to systems engineering," *Interacting with computers* (23:1) 2011, pp 4-17.

Beese, J., Haki, K., and Aier, S. "On the Conceptualization of Information Systems as Socio-Technical Phenomena in Simulation-Based Research,") 2015.

Bitzer, P., Söllner, M., and Leimeister, J.M. "Design Principles for High-Performance Blended Learning Services Delivery," *Business & Information Systems Engineering* (58:2) 2016, pp 135-149.

Blohm, I., Leimeister, J.M., and Krcmar, H. "Crowdsourcing: how to benefit from (too) many great ideas," *MIS Quarterly Executive* (12:4) 2013, pp 199-211.

Bonabeau, E. "Decisions 2.0: The power of collective intelligence," *MIT Sloan management review* (50:2) 2009, pp 45-52.

Davison, R. "GSS and action research in the Hong Kong police," *Information Technology & People* (14:1) 2001, pp 60-77.

Doan, A., Ramakrishnan, R., and Halevy, A.Y. "Crowdsourcing systems on the world-wide web," *Communications of the ACM* (54:4) 2011, pp 86-96.

Erickson, L.B., Trauth, E.M., and Petrick, I. "Getting inside your employees' heads: Navigating barriers to internal-crowdsourcing for product and service innovation," in: *Proceedings of the Thirty Third International Conference on Information Systems*, Orlando, FL, USA, 2012.

Fitzgerald, B., and Stol, K.-J. "The Dos and Dont's of Crowdsourcing Software Development," in: *SOFSEM 2015: Theory and Practice of Computer Science*, Springer, 2015, pp. 58-64.

Gaspoz, C. "Prediction Markets as Web 2.0 Tools for Enterprise 2.0," AMCIS, 2011.

Geiger, D., Rosemann, M., Fielt, E., and Schader, M. "Crowdsourcing information systems-definition, typology, and design,") 2012.

Geiger, D., and Schader, M. "Personalized task recommendation in crowdsourcing information systems—Current state of the art," *Decision Support Systems* (65) 2014, pp 3-16.

Giessmann, A., and Legner, C. "Designing business models for cloud platforms," *Information Systems Journal* (26:5) 2016, pp 551-579.

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

Hetmank, L. "Components and Functions of Crowdsourcing Systems-A Systematic Literature Review," *Wirtschaftsinformatik* (4) 2013.

Jette, A., Breck, A., and Johns, R. "Integrating Balanced Scorecard Performance Management with Crowdsourced Strategic Planning," Proceedings of the Transportation Research Board Annual Meeting (Washington DC), 2015.

Kohler, T., Fueller, J., Matzler, K., and Stieger, D. "Co-creation in virtual worlds: the design of the user experience," *MIS quarterly* (35:3) 2011, pp 773-788.

Kuek, S.C., Paradi-Guilford, C., Fayomi, T., Imaizumi, S., Ipeirotis, P., Pina, P., and Singh, M. "The global opportunity in online outsourcing," The World Bank.

Leavitt, H.J. "Applied Organisation Change in Industry: Structural, Technical and Human Approaches," *New Perspectives in Organisational Research, John Wiley*) 1964.

Leicht, N., Blohm, I., and Leimeister, J.M. "Leveraging the Power of the Crowd for Software Testing," *IEEE Software* (34:2) 2017, pp 62-69.

Leicht, N., Knop, N., Blohm, I., Müller-Bloch, C., and Leimeister, J.M. "When is Crowdsourcing Advantageous? The Case of Crowdsourced Software Testing,") 2016a.

Leicht, N., Rhyn, M., and Hansbauer, G. "Can Laymen Outperform Experts? The Effects of User Expertise and Task Design in Crowdsourced Software Testing," European Conference on Information Systems (ECIS 2016), Istanbul, Turkey, 2016b.

Lopez, M., Vukovic, M., and Laredo, J. "People Cloud Service for Enterprise Crowdsourcing," *Services - I, 2009 World Conference on*) 2010, pp 686-692.

Lyytinen, K., and Newman, M. "Explaining information systems change: a punctuated socio-technical change model," *European Journal of Information Systems* (17:6) 2008, pp 589-613.

Mumford, E. "The story of socio-technical design: Reflections on its successes, failures and potential," *Information Systems Journal* (16:4) 2006, pp 317-342.

Prpić, J., Shukla, P.P., Kietzmann, J.H., and McCarthy, I.P. "How to work a crowd: Developing crowd capital through crowdsourcing," *Business Horizons* (58:1) 2015, pp 77-85.

Rehman, F.U., Lbath, A., Murad, A., Rahman, M.A., Sadiq, B., Ahmad, A., Qamar, A., and Basalamah, S. "A Semantic Geo-Tagged Multimedia-Based Routing in a Crowdsourced Big Data Environment," Proceedings of the 23rd ACM international conference on Multimedia, ACM, 2015, pp. 759-760.

Rohrbeck, R., Thom, N., and Arnold, H. "IT tools for foresight: The integrated insight and response system of Deutsche Telekom Innovation Laboratories," *Technological Forecasting and Social Change* (97) 2015, pp 115-126.

Saxton, G.D., Oh, O., and Kishore, R. "Rules of crowdsourcing: Models, issues, and systems of control," *Information Systems Management* (30:1) 2013, pp 2-20.

Schenk, E., and Guittard, C. "Towards a characterization of crowdsourcing practices,") 2011.

Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R. "Action design research," *MIS quarterly*) 2011, pp 37-56.

Simic, K., Despotovic-Zrakic, M., Đuric, I., Milic, A., and Bogdanovic, N. "A MODEL OF SMART ENVIRONMENT FOR E-LEARNING BASED ON CROWDSOURCING," *RUO. Revija za Univerzalno Odlicnost* (4:1) 2015, p A1.

Simula, H. "The rise and fall of crowdsourcing?," System Sciences (HICSS), 2013 46th Hawaii International Conference on, IEEE, 2013, pp. 2783-2791.

Simula, H., and Vuori, M. "Benefits and barriers of crowdsourcing in B2B firms: Generating ideas with internal and external crowds," *International Journal of Innovation Management* (16:06) 2012, p 1240011.

Skopik, F., Schall, D., and Dustdar, S. "Discovering and managing social compositions in collaborative enterprise crowdsourcing systems," *International Journal of Cooperative Information Systems* (21:04) 2012, pp 297-341.

Soukhoroukova, A., Spann, M., and Skiera, B. "Sourcing, filtering, and evaluating new product ideas: An empirical exploration of the performance of idea markets," *Journal of Product Innovation Management* (29:1) 2012, pp 100-112.

Stocker, A., Richter, A., Hoefler, P., and Tochtermann, K. "Exploring appropriation of enterprise wikis," *Computer Supported Cooperative Work (CSCW)* (21:2-3) 2012, pp 317-356.

Street, C.T., and Meister, D.B. "Small business growth and internal transparency: The role of information systems," *MIS quarterly*) 2004, pp 473-506.

Sykes, T.A., Venkatesh, V., and Johnson, J.L. "Enterprise system implementation and employee job performance: Understanding the role of advice networks," *Mis Quarterly* (38:1) 2014, pp 51-72.

Thuan, N.H., Antunes, P., Johnstone, D., and Ha, X. "Building an enterprise ontology of business process crowdsourcing: a design science approach," *PACIS 2015 Proceedings*) 2015.

Tung, Y.-H., and Tseng, S.-S. "A novel approach to collaborative testing in a crowdsourcing environment," *Journal of Systems and Software* (86:8) 2013, pp 2143-2153.

Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., and Cleven, A. "Reconstructing the giant: On the importance of rigour in documenting the literature search process," ECIS, 2009, pp. 2206-2217.

Von Alan, R.H., March, S.T., Park, J., and Ram, S. "Design science in information systems research," *MIS quarterly* (28:1) 2004, pp 75-105.

Webster, J., and Watson, R.T. "Analyzing the past to prepare for the future: Writing a literature review," *Management Information Systems Quarterly* (26:2) 2002, p 3.

Weilbach, L., and Matthee, M.C. "Using the PSIC Model to Understand Change in an Educational Setting: The Case of an E-Textbook Implementation," ECIS, 2015.

Yan, M., Sun, H., and Liu, X. "iTest: testing software with mobile crowdsourcing," Proceedings of the 1st International Workshop on Crowd-based Software Development Methods and Technologies, ACM, 2014, pp. 19-24.

Zhu, H., Djurjagina, K., and Leker, J. "Innovative behaviour types and their influence on individual crowdsourcing performances," *International Journal of Innovation Management* (18:06) 2014, p 1440015.

Zogaj, S., and Bretschneider, U. "ANALYZING GOVERNANCE MECHANISMS FOR CROWDSOURCING INFORMATION SYSTEMS: A MULTIPLE CASE ANALYSIS,") 2014a.

Zogaj, S., Bretschneider, U., and Leimeister, J.M. "Managing crowdsourced software testing: a case study based insight on the challenges of a crowdsourcing intermediary," *Journal of Business Economics* (84:3) 2014b, pp 375-405.

Zogaj, S., Leicht, N., Bretschneider, U., Blohm, I., and Leimeister, J.M. "Towards Successful Crowdsourcing Projects: Evaluating the Implementation of Governance Mechanisms,") 2015.

Zuchowski, O., Posegga, O., Schlagwein, D., and Fischbach, K. "Internal crowdsourcing: conceptual framework, structured review, and research agenda," *Journal of Information Technology* (31:2) 2016, pp 166-184.