Please quote as: Leicht, N.; Blohm, I. & Leimeister, J. M. (2016): How to Systematically Conduct Crowdsourced Software Testing? Insights from an Action Research Project. In: International Conference on Information Systems (ICIS), Dublin, Ireland.

How to Systematically Conduct Crowdsourced Software Testing? Insights from an Action Research Project

Research-in-Progress

Niklas Leicht[1]
niklas.leicht@unisg.ch

Ivo Blohm[1]
ivo.blohm@unisg.ch

Jan Marco Leimeister[1,2]
janmarco.leimeister@unisg.ch

[1]University of St. Gallen
Institute of Information Management
Unterer Graben 21, CH-9000 St. Gallen, Switzerland

[2]University of Kassel
Research Center for IS Design (ITeG)
Pfannkuchstr. 1, D-34121 Kassel, Germany

Abstract

Nowadays, traditional testing approaches become less feasible – both economically and practicably - for several reasons, such as an increasingly dynamic environment, shorter product lifecycles, cost pressure, as well as a fast growing and increasingly segmented hardware market. With the surge towards new modes of value creation, crowdsourced software testing (CST) seems to be a promising solution to effectively solve these problems and was already applied in various software testing contexts. However, literature so far mostly neglected the perspective of an organization intending to crowdsource tasks. In this study, we present an ongoing action research project with a consortium of six companies and present a preliminary model for crowdsourced software testing in organizations. The model unfolds necessary activities, process changes, and the accompanied roles for crowdsourced software testing to enable organizations to systematically conduct such initiatives and illustrates how test departments can use crowdsourcing as a new tool in their department.

Keywords: Crowdsourcing, software testing, action research, IS development

Introduction

Software testing is an integral part to ensure quality in software development. However, many IT departments face an increasingly dynamic environment, shorter product lifecycles and cost pressure. On top of that, the rapid development of new IT-enabled business models and a fast growing hardware market as well as its segmentation, that is, smartphones, tablets, wearable technologies or the "Internet of things", lead to an increasing degree of complexity in software testing. Given the increased complexity, the domain of software testing is about to develop manifold approaches to overcome this issue. One approach is test automation, i.e., the automated execution of pre-scripted tests via software (Huizinga and Kolawa 2007). However, since automated testing is still not applicable in many settings (Rafi et al. 2012) and most tasks still require human intelligence to be performed, traditional approaches to software testing are becoming less applicable – both economically and practicably (Bacon et al. 2009; Dolstra et al. 2013). With the surge toward collective undertaking of production processes in the context of new information and communication technologies (Boehm 2006; Tajedin and Nevo 2013), the concept of crowdsourcing – that is, the proposal of activities to a group of individuals who voluntarily undertake tasks based on a flexible open call (Blohm et al. 2013) – has gained track to effectively solve business problems (Brabham 2008; Jeppesen and Lakhani 2010) and recently found its application in software testing (LaToza and van der Hoek 2016). In crowdsourced software testing (CST) or crowdtesting, a diverse pool of people test software in real environments using their own devices leveraging the "wisdom of the crowds" (Surowiecki 2005) or as Linus' Law, one of the mantras of open source movement, states: "Given enough eyeballs, all bugs are shallow" (Raymond 1999).

While the research on this topic is still in its inception, a number of studies have already examined some of the implications of CST. Existing literature predominantly describes crowdsourcing as a system with three components, the crowd, the intermediary, and the IT platform. CST was applied for the testing of graphical user interfaces and mobile applications and was found to be both technically feasible and sufficiently reliable (Amini et al. 2012; Dolstra et al. 2013; Liu et al. 2012). Furthermore, Hossfeld et al. (2014) assess key issues in crowdsourced user experience testing and provide a collection of best practices to address the challenges in crowdtesting projects (Stol and Fitzgerald 2014). From a system's perspective, Bacon et al. (2009) provide a comprehensive overview of the application of crowdsourcing in bug and vulnerability reporting. These studies deliver first insights that crowdsourcing can be a feasible option in software testing in different scenarios. Building on that, there are first attempts to identify specific scenarios in which CST might be a superior mode of creation compared to in-house testing (Leicht et al. 2016a). Moreover, there are first findings on how task design, expertise (Leicht et al. 2016b), and other factors such as process support and guidance (Tung and Tseng 2013) or time constraints (Mäntylä and Itkonen 2013) positively influence the performance of the crowd. The crowd is usually reached through an intermediary who has three main challenges to tackle: managing the settlement process, managing the crowd, and managing the technology (Tajedin and Nevo 2013; Zogaj et al. 2014).

Despite these merits, the perspective of an organization that seeks to crowdsource task has largely been neglected in crowdsourced software testing research, and is only examined on a conceptual level (Afuah and Tucci 2012; Estellés-Arolas and González-Ladrón-de-Guevara 2012). Thus, we follow the call of various researchers (e.g., Leicht et al. 2015; Zhao and Zhu 2014) and investigate crowdsourcing in software testing from an organization's point of view. Accordingly, this paper aims to answer the following research question: How to use crowdsourcing and systematically conduct crowdsourced software testing initiatives? To answer this question, we report an ongoing action research project with a consortium of six companies that desire to employ crowdsourced software testing to uncover the necessary activities and tasks to be performed to leverage CST in their department. As a result, we present a preliminary model for mediated crowdsourcing. Mediated crowdsourcing is a popular form of crowdsourcing in the field of software testing where an intermediary connects organizations that want to apply crowdsourcing with potential crowdsourcees and manage the crowdsourcing process (Ipeirotis 2010; Zogaj et al. 2014). Our expected contribution is twofold. First, by investigating crowdsourcing from a seeker's perspective we help to synthesize crowdsourcing literature that so far mostly neglected this perspective. Second, the developed process model structures the CST process and unfolds necessary activities as well as key levers in the context of software testing. Thus, we create deeper insights into the mechanisms of crowdsourcing in a software testing context. For practitioners, the paper illustrates how test departments can use CST and provides a model as a guideline with precise action taking in software testing projects for managers.

The remainder of the paper proceeds as follows: First, section 2 outlines the conceptual background of our research and introduces crowdsourced software testing as an application of crowdsourcing in practice. Section 3 is dedicated to our research approach and the project setting of our action research project. Following that, we report the first research cycle. Fourth, we give an overview of the planned second cycle. Last, we discuss expected contributions from our research as well as implications for practitioners, such as test managers and executives.

Related Work

Crowdsourcing

Crowdsourcing represents a new form of value creation and requires different activities and responsibilities compared to in-house approaches or outsourcing (Bernstein et al. 2012). Compared to outsourced testing activities, crowdsourcing represents a fundamental shift and extends the frontiers of available skills or knowledge beyond the boundaries of organizations or suppliers (Geiger et al. 2012). With crowdsourcing, it is possible to mobilize the expertise and creativity distributed among a large panel of people in order to achieve a certain set of tasks (Schenk and Guittard 2011), and not just preselected testers from a supplier (Afuah and Tucci 2012). However, the roles and activities in crowdsourcing can also vary. Figure 1 depicts the different models of crowdsourcing compared to traditional outsourcing. In traditional outsourcing, the principal negotiates the terms of the outsourcing degree, as well as the mode of outsourcing and the outsourcing company delivers the a priori specified services (Dibbern et al. 2004). In mediated crowdsourcing (II), an organization selects an intermediary who acts as a service provider. The intermediary mainly manages the crowdsourcing process including the identification and invitation of the crowdsourcees, the allocation of tasks, as well as the remuneration for the crowdsourcees via its IT-based platform (Blohm et al. 2016; Zogaj et al. 2014). The tasks are then processed by the crowdsourcees on the intermediary's IT platform. The organization usually pays a fixed fee for the use of the service and interacts only directly with managers from the intermediary. Of course, an organization can choose to perform crowdsourcing without an intermediary in place (III). In this scenario, the organization takes the responsibilities of the intermediary and has to perform various other tasks and governance mechanisms, such as the motivation and incentives of crowdsourcees, communication with crowdsourcees, and many other (see Zogaj et al. 2015 for details).

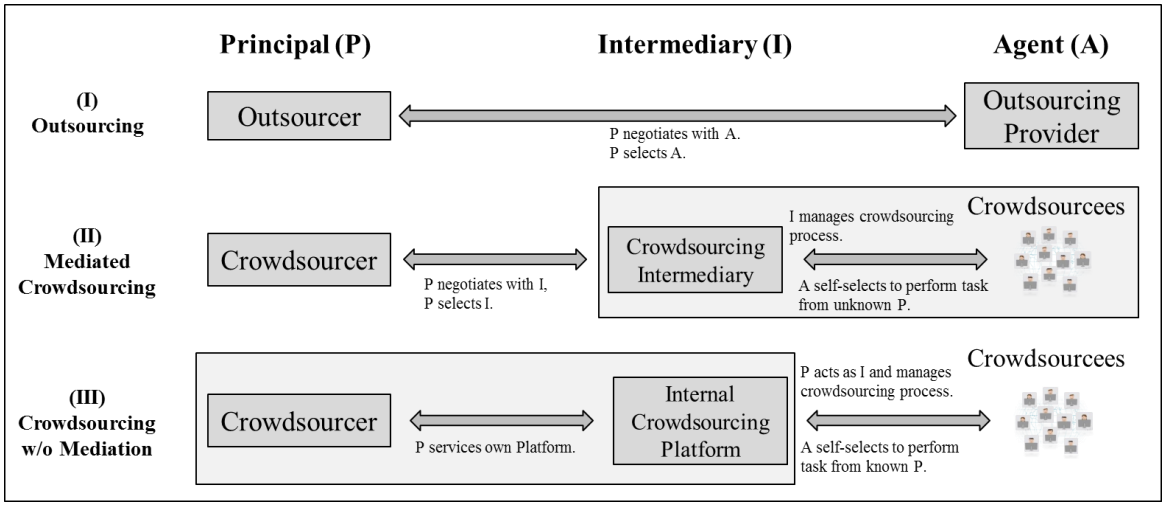

Figure 1. Sourcing Forms (adapted from: Hoßfeld et al. 2012; Zogaj et al. 2014)

Crowdsourced Software Testing

Crowdsourced software testing (CST) or crowdtesting is a specific application of crowdsourcing in the domain of software development. It refers to the outsourcing of software testing activities to the crowd. It grants access to a diverse pool of people who voluntarily test software in real environments using their own devices (Leicht et al. 2016a; Zogaj et al. 2014). We define software testing as a verification process for the

assessment of software quality and a process for achieving that quality by supporting the interests of all stakeholders of an application, that is, end-users, developers, software designers, and software testers (Bertolino 2007; Naik and Tripathy 2011). To achieve that goal, it becomes clear that different types of testing by various stakeholders at different times (during or subsequent to development) need to be done (Roggio et al. 2013). That means, that software testing in this sense is much more than just "the process of executing a program with the intent of finding errors" (Myers et al. 2011, p. 6).

CST can be applied in numerous types of testing, but research so far usually applied CST in dynamic testing scenarios where a written code is executed and examined by the crowd. Further, the crowd is mostly concerned with output given specific inputs since they do not know or see the source code. This is referred to as black box testing (or "end-user testing") where the software itself is seen as a blackbox and only inputs and outputs are visible for the tester. This form of testing not only covers functional aspects, it also covers non-functional aspects such as performance, reliability, and security of a software (Roggio et al. 2013). The most common type of testing where CST gained increasing popularity is verification, validation and acceptance testing. While verification testing aims to eliminate defects and faults that cause error states (functional black-box testing), in validation testing the end-user runs tests to determine if specific inputs result in specific outputs to ensure no failures are experienced (Stutzke 2005). Another important part of software testing is concerned with the usability of a software that is defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO 9241-11). Subsequently, usability testing is the test of software with the goal of improving it (Dumas and Redish 1999). In this vein, beta testing can be considered as both validation and acceptance, as well as usability testing depending on its goal. Table 1 gives an overview of the application of CST in different types of testing.

Table 1. Overview of Application of CST Research

Type of Testing	Authors
Functional and Verification Testing	Dolstra et al. (2013); Leicht et al. (2016a); Mäntylä and Itkonen (2013); Chen and Luo (2014)
Non-Functional Testing	Chen and Luo (2014) (performance testing)
Validation and Acceptance Testing	Amini et al. (2012); Leicht et al. (2016a); Tung and Tseng (2013);
Usability Testing	Hoßfeld et al. (2014); Liu et al. (2012)

As with other crowdsourcing applications, companies, meaning the crowdsourcer, can either directly interact with the crowd or they can use intermediaries who provide this service for a fee (Chanal and Caron-Fasan 2010). These intermediaries act as brokers who connect the organizations that want to apply crowdsourcing with potential crowdsourcees and manage the crowdsourcing process (Ipeirotis 2010; Zogaj et al. 2014). Depending on the type of testing (e.g., (non-)functional testing, usability testing), these tasks as well as the targeted crowds can be very diverse (Stol and Fitzgerald 2014).

Methodology

Action Research

In order to study crowdsourced software testing initiatives in an organizational context and to derive a process model, we applied action research. Action research is future oriented and does not strive for distanced and generalizable explanations or the prediction of coherences, but the joint understanding and learning by researchers and subjects as well as the changing of actual conditions based on a real problem (Baskerville and Myers 2004; Susman and Evered 1978). Thus, the essential purpose of action research is to address concerns to individuals or organizations with the broader purpose to contribute to the solution of this concern (Reason 2006). Action research combines the theoretical knowledge of the researchers with the subject's practical and situated insights and has established as viable method in information systems research in general (e.g., Kohler et al. 2011; Lindgren et al. 2004; Puhakainen and Siponen 2010) and also for implementing crowdsourcing mechanisms in organizations (Haas et al. 2015). It is especially suited when the researcher needs to become deeply involved with the subject's problem and environment (Füller

et al. 2011; Street and Meister 2004). To invoke the deep involvement, action research is a cyclic and multiphase process consisting of five iterative phases: (1) diagnosing, (2) action planning, (3) action taking, (4) evaluating and (5) specifying learning (Aguinis 1993; Baskerville and Wood-Harper 1996). However, because the researchers are directly involved in the project, action research is about choices. The quality of action research rests on the ability of the researchers to see the choices that are made and understand the consequences, as well as to make the choices transparent and communicate them to a wider public (Reason 2006).

Project Setting

The research project is built on the basic principles of consortium research (Österle and Otto 2010). Consortium research aims to develop solutions for a problem class within a collaborative environment. It profits from the collaboration and exchange of expertise from researchers as well as partner companies who all share the same goal. The process is iterative in its nature and thereby fits into the action research approach with two cycles. To identify how to profit from crowdsourced software testing, we started the initiative with six companies that monitored CST for a while but had no experience with crowdsourcing so far and subsequently struggled to use this new form of software testing in their departments. These companies all believe that testing with the crowd can help them to improve on the overall testing performance. To tackle this issue an interdisciplinary project team was composed consisting of researchers specialized in crowdsourcing and several experts in software testing, that is, test managers, head of testing services, and testers from the involved companies.

Research Cycles

In order to explore the mechanisms of crowdsourced software testing and develop a systematic approach, we intend to conduct two action research cycles. For the first cycle, we chose to investigate mediated CST. The reason we chose this setting for our first cycle is twofold. First, this setting represents a very common scenario in crowdsourced software testing where an intermediary acts as a service provider and is managing the crowdsourcing process. Hence, the test manager of the organization is only communicating with the intermediary who manages the crowd and the testing process. Second, since companies are accustomed to having software testing performed externally, the shift to using mediated crowdsourcing becomes less complex but at the same time unveils key insights on the differences between traditional sourcing and crowdsourcing. As for the second cycle, we intend to expand the model for crowdsourcing without mediation that is more complex in its nature because the organization has to perform several activities performed by the intermediary in the mediated CST approach.

Action Research Cycle One

Phase 1: Diagnosing

With the intent of creating an approach to systematically conduct crowdsourced software testing, we first investigated the existing testing processes and practices. We monitored the testing process in the companies and reviewed documentation to gain a deeper understanding of existing practices. Further, we wanted to develop a deeper understanding why test managers did not apply CST so far. For this reason we conducted exploratory interviews with the involved test managers (N=3). Moreover, we interviewed three intermediaries who are all specialized in different kinds of testing (e.g., usability testing, functional testing) to understand the mediation process and the requirements to a company from an intermediary's perspective. Our interviewees (N=3) were experienced test managers who execute and manage crowdtests with different clients on a regular basis. Finally, to ground the diagnosis in theory and further enhance our understanding, we reviewed existing literature regarding these matters. This triangulation of data allowed us to derive pivotal requirements and challenges to be addressed by our solution.

The interviews with the test managers revealed that they either did not know how to frame a test to use crowdsourced testing but also were not sure what tasks they have to perform in mediated crowdsourcing. Statements from managers of intermediaries supported this by stating that many customers were unsure about their objective and the test scope and that these two activities are vital to project success. Another concern by the test experts was, that the defect management process could become excessively extensive

due to many submissions of low quality (Ipeirotis and Paritosh 2011) or duplicate bug reports. Furthermore, the diagnosis revealed that the involvement for test managers in the test itself and defect management are different in CST. Although the intermediary communicates with the crowd, there are often specific issues where support from the companies' test manager is needed during the tests.

Besides that, security and intellectual property is another issue in crowdsourced software testing (Leicht et al. 2016a; Stol and Fitzgerald 2014). By opening to a large, anonymous crowd, companies are afraid that crowdsourcees could, instead of testing the software, try to hack the application and harm the company. In this regard, the test managers were also concerned about industrial spying, as well as public exposure due to low software quality of the test object. This leads to the conclusion that it is of high importance to check if crowdsourcing can be applied taking IT and data security regulations in account. Moreover, test managers need to carefully consider in which software testing phase to use crowdsourcing and how to scope the test.

Phase 2: Action Planning

Based on the diagnosis and informed by standard project management literature (Cleland and Ireland 1994; Wysocki 2011) we framed the CST initiative as a small sub-project in the testing process and created a four-phase process (cf. figure 2). As a next step we attempted to address the expressed concerns and necessary steps and built high-level activities within every phase.

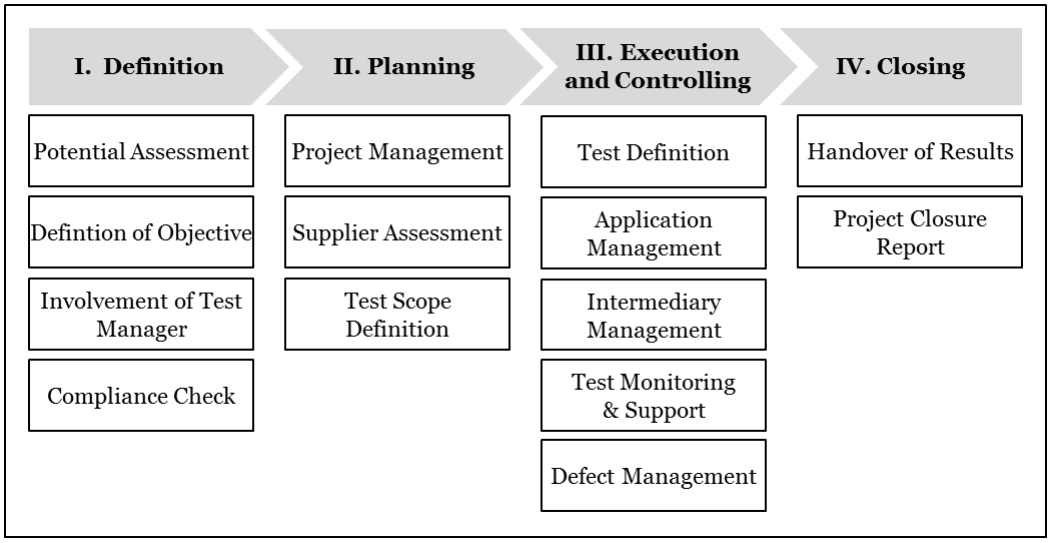

Figure 2. Preliminary CST Model

This high-level model was elaborated and further specified in two expert workshops with test managers of the involved companies as well as in single interviews with the test managers (N=4) and ultimately led to the description of 14 activities in mediated crowdsourced software testing. Table 2 explains the particular activities for each phase.

Table 2. Overview of Activities

Phase	Activity	Description
I. Definition	1. Potential Assessment	Identification and assessment of potential for the application of CST for the specific project, i.e., software object.
	2. Definition of Objective	Definition of test objective. Crowdsourced software testing can be leveraged to accomplish several testing goals, such as functionality, usability, non-functional aspects, reliability etc.
	3. Involvement of Test Manager	Depending on the objectives, the test manager can select between various service levels varying in the degree of involvement. Also, the crowdsourcees are defined, i.e., external crowdsourcees, customers, or employees.

	4. Compliance Check	Ensuring that the project does not violate any regulatory or compliance guidelines. This mainly focuses on data and IT security but may also affect other departments (e.g., human resources for testing with employees)
II. Planning	5. Project Management	Setup of a basic project management office (PMO) to coordinate all relevant CST aspects, i.e., time schedule for crowdtests, budget controlling, and definition of roles.
	6. Supplier Assessment	Assessment and selection of an intermediary for the execution of the crowdtest depending on the previously defined parameters such as objective, crowd and involvement.
	7. Test Scope Definition	Definition of the type (functional, usability, etc.) and extent of the test (entire software vs. functions), the specific definition of crowdsourcees, as well as the integration into the overarching test concept.
III. Execution and Controlling	8. Test Definition	Detailed design of test cases and test instructions as well as the definition of reporting formats.
	9. Application Management	Depending on the objective, the software to be tested has to fulfill certain quality criteria. The test manager has to ensure the maturity of the software before the crowdtest.
	10. Intermediary Management	Settlement of the responsibilities for the test and the examination of results. Also, necessary information for the test and an exact schedule for the tests has to be provided and set.
	11. Test Monitoring & Support	Depending on the chosen service level (3), the test manager has to be available on demand and provide support if necessary.
	12. Defect Management	Verification, categorization, and prioritization of submitted bugs/ examination of qualitative crowdsourcee feedback and integration of the results into the overall test results.
IV. Closing	13. Handover of Results	After completion, the test manager delivers the results back to the project team. The handover document contains the results of the test, i.e., verified bug reports and notes for possible change requests. Further, the test manager elaborates in how far the objectives set in step two were fulfilled with the crowdtest.
	14. Project Closure Report	The project closure report contains a summary of the results as well as lessons learned. To ensure continuous improvement, the lessons learned should identify process improvements and operationalize concrete action taking.

Phase 3: Action Taking

The project setting allowed us to conduct a crowdsourced software testing project with the developed model and its activities in three different companies. The collaboration and ongoing interaction with the respective test managers proved especially helpful to collect valuable feedback and learnings as the consortium approach enabled a very fruitful interaction between all involved parties. Table 3 depicts the involved companies as well as the test object and the main objective of the test. To enhance generalizability the tests were conducted with different intermediaries per test. To further enhance the generalizability of the results we chose to apply the model in both development paradigms and with different test objects. In addition, the type of testing varied between the companies. The tests itself were conducted by 20-30 crowdsourcees for functional testing and 15 for usability testing. The duration of the tests was about 48-72 hours in each scenario.

Table 3. Test Settings			
Company	**Development Paradigm**	**Test Object**	**Type of Testing**
Bank	Waterfall	Mobile Banking Application	Functionality
Insurance Company	Agile	Corporate Public Website	Usability
Online Retailer	Agile	Online Shop (Web)	Functionality & Usability

Planned Phase 4: Evaluating

For the first cycle, we focused on a formative evaluation of the project. A formative evaluation is especially useful for empirical interpretations that serve as a basis for the improvement of the developed model (Venable et al. 2016). To evaluate and improve the process steps, we interviewed the involved test and project managers (N=6) before and after the project. For these interviews, we created a roughly structured interview guideline with questions regarding every phase of the crowdtest, the overall satisfaction with the process, the test results, and the collaboration with the intermediary. The participants should especially elaborate on aspects that caused disturbance or were perceived as not timely, i.e., too early or too late in the process. All interviews were recorded and subsequently transcribed. In addition, detailed notes were taken during the interviews. Moreover, we conducted a focus group workshop (Morgan 1996) with the consortium (N=13) to identify pitfalls and improvements of every phase for the second research cycle and to synthesize the knowledge from all initiatives, which will determine the learning in the next phase.

Planned Phase 5: Specifying Learning

Although our evaluation is a work-in-progress, first findings indicate that crowdsourced software testing with an intermediary requires a different workflow for the test manager. In contrast to outsourcing or in-house testing where the manager receives the results from professionals, crowdsourced software testing is often performed by semi-professional or even untrained testers. This fact leads to an increased effort in defect management. Another fact that intensifies the efforts is that in most cases, crowdsourcees are not testing against a specification and the intermediary usually also does not have a specification of the software at hand. These circumstances produce bugs professional testers would not have submitted and, in conclusion, lead to an increased effort in defect management. Also, the input of other stakeholders, for example compliance or security department, is needed and the examination process often requires time and additional action taking by the manager. This is especially true if the organization has no previous experience with crowdsourcing. Hence, the findings suggest that a new project role, the "crowdtest manager", should be established, especially in larger projects with different development and test streams.

Planned Action Research Cycle Two

As for the second research cycle, we intend to apply and further develop the preliminary model from the first cycle for settings where the intermediation is performed by the crowdsourcing company itself. In this scenario, complexity is higher because the company has to perform additional activities once performed by the intermediary, such as, crowd management or remuneration of the crowdsourcees. For this purpose, we also intend to apply the model in three real-world testing projects of our consortium. Table 4 summarizes our approach for the second cycle.

Table 4. Planned Cycle Two	
Phase	**Planned Actions**
Diagnosing	In the first cycle, the intermediary is seen as a "black box" performing a variety of important tasks and governance mechanisms to enable crowdsourced software testing. Building on literature (e.g., Zogaj et al. 2014; Zogaj et al. 2015), we intend to "white box" the intermediary and clarify which tasks have to be performed by the organization. Further, exploratory interviews with managers from crowdsourcing intermediaries will be conducted to shed light on that issue.
Action Planning	Based on the learnings from the first cycle, the model will be further developed and enhanced to be applicable for scenarios without intermediation as well as hybrid forms. That includes necessary changes in existing activities, but mainly the inclusion of activities performed by an intermediary such as crowd and community management, and others. Further, the results revealed that existing roles in the testing process must be adjusted and new roles, e.g., a crowd manager responsible for the communication with the crowd need to be considered and employed.
Action Taking	The adapted model including the specified learning will then again be applied in three different companies with different scenarios to ensure generalizability. We intend to apply the model for internal CST with employees where only the platform is provided, as well as completely without mediation with customers of the company as a crowd.
Evaluation	In order to evaluate the practicability and overall performance of the developed model, we intend to apply a summative evaluation approach. Summative evaluations focus on meaning an support decisions regarding the applicability of the model (Venable et al. 2016). We will conduct semi-structured interviews with the test managers as well as the project managers with a focus on the real world usefulness and realized benefits of the applied CST approach.
Specified Learning	The model developed in the action research project intends to provide an exhaustive and complete guideline on how to conduct CST initiatives and thereby claims to be applicable for a variety of different crowdsourcing scenarios in software testing. Depending on the results of the evaluation, we will decide if these criteria are met or a third action cycle should be conducted.

Expected Contribution and Future Research

To the best of our knowledge, this paper is the first to investigate crowdsourced software testing from an organizational point of view. Traditional software testing, performed in-house or outsourced, heavily relies on the knowledge of specific and very few people with a limited locus and an ex-ante selection of results. Crowdsourcing flips this concept by broadening the locus due to the open nature of the call and follows the mantra of open source code, which has proven to be a feasible concept over the last decades. This study will help to create deeper insights into the mechanisms of crowdsourcing from an organization's point of view. Thus, we help to synthesize literature of crowdsourcing that so far mostly focused on the perspective of the crowdsourcee or intermediary.

Preliminary results reveal that, although the crowdsourcing process itself is managed by an intermediary, crowdsourcing still requires other process structures and tasks to be performed than traditional outsourcing of test activities. Hence, the developed model unfolds necessary changes regarding process design and the accompanied roles in the test process with the use of crowdsourced software testing. Further, the expected model identifies activities that are of particular importance for the success of CST initiatives. However, we argue that our results are not just limited to CST. In fact, due to the complex nature of the task and the modularizability as well as the interdependence of activities in software development, we propose that our results are transferable to other activities in the development process, that is, code production or even earlier stages of development such as design evaluation and even requirements elicitation. For practice, the paper illustrates how test departments can utilize crowdsourcing as a new tool in their collection. On top of that, we expect to provide a process model as a guideline with precise action taking in software testing projects for test managers to enable crowdsourced software testing in testing departments.

References

Afuah, A., and Tucci, C.L. 2012. "Crowdsourcing as a Solution to Distant Search," *Academy of Management Review* (37:3), pp. 355-375.

Aguinis, H. 1993. "Action Research and Scientific Method: Presumed Discrepancies and Actual Similarities," *The Journal of applied behavioral science* (29:4), pp. 416-431.

Amini, S., Lin, J., Hong, J., Lindqvist, J., and Zhang, J. 2012. "Towards Scalable Evaluation of Mobile Applications through Crowdsourcing and Automation," in: *CMU-CyLab-12-006, Carnegie Mellon University*.

Bacon, D.F., Chen, Y., Parkes, D., and Rao, M. 2009. "A Market-Based Approach to Software Evolution," in: *ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*. ACM, pp. 973-980.

Baskerville, R., and Myers, M.D. 2004. "Special Issue on Action Research in Informaton Systems: Making Is Research Relevant to Practice Foreword," *MIS Quarterly* (28:3), p. 2.

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

Bernstein, A., Klein, M., and Malone, T.W. 2012. "Programming the Global Brain," *Communications of the ACM* (55:5), pp. 41-43.

Bertolino, A. 2007. "Software Testing Research: Achievements, Challenges, Dreams," *2007 Future of Software Engineering*: IEEE Computer Society, pp. 85-103.

Blohm, I., Leimeister, J.M., and Krcmar, H. 2013. "Crowdsourcing: How to Benefit from (Too) Many Great Ideas," *MIS Quarterly Executive* (4:12), pp. 199-211.

Blohm, I., Riedl, C., Füller, J., and Leimeister, J.M. 2016. "Rate or Trade? Identifying Winning Ideas in Open Idea Sourcing," *Information Systems Research* (27:1), pp. 27-48.

Boehm, B. 2006. "A View of 20th and 21st Century Software Engineering," in: *International conference on Software engineering (ICSE 2006)*. ACM, pp. 12-29.

Brabham, D.C. 2008. "Crowdsourcing as a Model of Problem Solving," *Convergence: The International Journal of Research into New Media Technologies* (14:1), pp. 75-90.

Chanal, V., and Caron-Fasan, M.-L. 2010. "The Difficulties Involved in Developing Business Models Open to Innovation Communities: The Case of a Crowdsourcing Platform," *M@n@gement* (13:4), pp. 318-340.

Chen, Z., and Luo, B. 2014. "Quasi-Crowdsourcing Testing for Educational Projects," in: *International Conference on Software Engineering (ICSE 2014)*. ACM, pp. 272-275.

Cleland, D.I., and Ireland, L.R. 1994. *Project Management: Strategic Design and Implementation*. McGraw-Hill New York, NY.

Dibbern, J., Goles, T., Hirschheim, R., and Jayatilaka, B. 2004. "Information Systems Outsourcing: A Survey and Analysis of the Literature," *ACM SIGMIS Database* (35:4), pp. 6-102.

Dolstra, E., Vliegendhart, R., and Pouwelse, J. 2013. "Crowdsourcing Gui Tests," in: *Sixth International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, pp. 332-341.

Dumas, J.S., and Redish, J. 1999. *A Practical Guide to Usability Testing*. Intellect Books.

Estellés-Arolas, E., and González-Ladrón-de-Guevara, F. 2012. "Towards an Integrated Crowdsourcing Definition," *Journal of Information Science* (38:2), pp. 189-200.

Füller, J., Hutter, K., and Faullant, R. 2011. "Why Co-Creation Experience Matters? Creative Experience and Its Impact on the Quantity and Quality of Creative Contributions," *R&D Management* (41:3), pp. 259-273.

Geiger, D., Rosemann, M., Fielt, E., and Schader, M. 2012. "Crowdsourcing Information Systems - Definition Typology, and Design," in: *International Conference on Information Systems (ICIS 2012)*. Orlando, USA.

Haas, P., Blohm, I., Peters, C., and Leimeister, J.M. 2015. "Modularization of Crowdfunding Services – Designing Disruptive Innovations in the Banking Industry," *International Conference on Information Systems (ICIS 2015)*, Austin, USA.

Hoßfeld, T., Hirth, M., and Tran-Gia, P. 2012. "Crowdsourcing," *Informatik-Spektrum*), pp. 1-5.

Hoßfeld, T., Keimel, C., Hirth, M., Gardlo, B., Habigt, J., Diepold, K., and Tran-Gia, P. 2014. "Best Practices for Qoe Crowdtesting: Qoe Assessment with Crowdsourcing," *IEEE Transactions on Multimedia* (16:2), pp. 541-558.

Huizinga, D., and Kolawa, A. 2007. *Automated Defect Prevention: Best Practices in Software Management*. John Wiley & Sons.

Ipeirotis, P.G. 2010. "Analyzing the Amazon Mechanical Turk Marketplace," *XRDS: Crossroads, The ACM Magazine for Students* (17:2), pp. 16-21.

Ipeirotis, P.G., and Paritosh, P.K. 2011. "Managing Crowdsourced Human Computation: A Tutorial," *Proceedings of the 20th international conference companion on World wide web*: ACM, pp. 287-288.

ISO 9241-11. 1998. *Ergonomics Requirements for Office with Visual Display Terminals (Vdts)*. Geneva: International Organization for Standardization.

Jeppesen, L.B., and Lakhani, K.R. 2010. "Marginality and Problem-Solving Effectiveness in Broadcast Search," *Organization Science* (21:5), Sep-Oct, pp. 1016-1033.

Kohler, T., Fueller, J., Matzler, K., and Stieger, D. 2011. "Co-Creation in Virtual Worlds: The Design of the User Experience," *MIS Quarterly* (35:3), pp. 773-788.

LaToza, T.D., and van der Hoek, A. 2016. "Crowdsourcing in Software Engineering: Models, Motivations, and Challenges," *IEEE software* (33:1), pp. 74-80.

Leicht, N., Durward, D., Blohm, I., and Leimeister, J.M. 2015. "Crowdsourcing in Software Development: A State-of-the-Art Analysis," in: *28th Bled eConference*. Bled, Slovenia.

Leicht, N., Knop, N., Blohm, I., Müller-Bloch, C., and Leimeister, J.M. 2016a. "When Is Crowdsourcing Advantageous? The Case of Crowdsourced Software Testing," in: *European Conference on Information Systems (ECIS 2016)*. Istanbul, Turkey.

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

Lindgren, R., Henfridsson, O., and Schultze, U. 2004. "Design Principles for Competence Management Systems: A Synthesis of an Action Research Study," *MIS Quarterly* (28:3), pp. 435-472.

Liu, D., Bias, R.G., Lease, M., and Kuipers, R. 2012. "Crowdsourcing for Usability Testing," *Proceedings of the American Society for Information Science and Technology* (49:1), pp. 1-10.

Mäntylä, M.V., and Itkonen, J. 2013. "More Testers–the Effect of Crowd Size and Time Restriction in Software Testing," *Information and Software Technology* (55:6), pp. 986-1003.

Morgan, D.L. 1996. *Focus Groups as Qualitative Research*. Sage publications.

Myers, G.J., Sandler, C., and Badgett, T. 2011. *The Art of Software Testing*. John Wiley & Sons.

Naik, K., and Tripathy, P. 2011. *Software Testing and Quality Assurance: Theory and Practice*. John Wiley & Sons.

Österle, H., and Otto, B. 2010. "Consortium Research," *Business & Information Systems Engineering* (2:5), pp. 283-293.

Puhakainen, P., and Siponen, M. 2010. "Improving Employees' Compliance through Information Systems Security Training: An Action Research Study," *MIS Quarterly* (34:4), pp. 757-778.

Rafi, D.M., Moses, K.R.K., Petersen, K., and Mäntylä, M.V. 2012. "Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey," in: *Proceedings of the 7th International Workshop on Automation of Software Test*. IEEE, pp. 36-42.

Raymond, E. 1999. "The Cathedral and the Bazaar," *Knowledge, Technology & Policy* (12:3), pp. 23-49.

Reason, P. 2006. "Choice and Quality in Action Research Practice," *Journal of Management Inquiry* (15:2), pp. 187-203.

Roggio, R.F., Gordon, J.S., and Comer, J.R. 2013. "Taxonomy of Common Software Testing Terminology: Framework for Key Software Engineering Testing Concepts," in: *Proceedings of the Conference for Information Systems Applied Research*. p. 1508.

Schenk, E., and Guittard, C. 2011. "Towards a Characterization of Crowdsourcing Practices," *Journal of Innovation Economics & Management*:1), pp. 93-107.

Stol, K.-J., and Fitzgerald, B. 2014. "Two's Company, Three's a Crowd: A Case Study of Crowdsourcing Software Development," in: *International Conference on Software Engineering (ICSE 2014)*. pp. 187-198.

Street, C.T., and Meister, D.B. 2004. "Small Business Growth and Internal Transparency: The Role of Information Systems," *MIS Quarterly* (28:3), pp. 473-506.

Stutzke, R.D. 2005. *Estimating Software-Intensive Systems: Projects, Products, and Processes*. Pearson Education.

Surowiecki, J. 2005. *The Wisdom of Crowds*. Random House LLC.

Susman, G.I., and Evered, R.D. 1978. "An Assessment of the Scientific Merits of Action Research," *Administrative Science Quarterly*), pp. 582-603.

Tajedin, H., and Nevo, D. 2013. "Determinants of Success in Crowdsourcing Software Development," in: *Proceedings of the 2013 annual conference on Computers and people research*. ACM, pp. 173-178.

Tung, Y.-H., and Tseng, S.-S. 2013. "A Novel Approach to Collaborative Testing in a Crowdsourcing Environment," *Journal of Systems and Software* (86:8), pp. 2143-2153.

Venable, J., Pries-Heje, J., and Baskerville, R. 2016. "Feds: A Framework for Evaluation in Design Science Research," *European Journal of Information Systems* (25:1), pp. 77-89.

Wysocki, R.K. 2011. *Effective Project Management: Traditional, Agile, Extreme*. John Wiley & Sons.

Zhao, Y., and Zhu, Q. 2014. "Evaluation on Crowdsourcing Research: Current Status and Future Direction," *Information Systems Frontiers* (16:3), pp. 417-434.

Zogaj, S., Bretschneider, U., and Leimeister, J.M. 2014. "Managing Crowdsourced Software Testing: A Case Study Based Insight on the Challenges of a Crowdsourcing Intermediary," *Journal of Business Economics* (84:3), pp. 375-405.

Zogaj, S., Leicht, N., Bretschneider, U., Blohm, I., and Leimeister, J.M. 2015. "Towards Successful Crowdsourcing Projects: Evaluating the Implementation of Governance Mechanisms," in: *International Conference on Information Systems (ICIS 2015)*. Fort Worth, USA.