28th Bled eConference

#eWellBeing

June 7 - 10, 2015; Bled, Slovenia

Crowdsourcing in Software Development: *A State-of-the-Art Analysis*

Niklas Leicht

University of St.Gallen, Switzerland

niklas.leicht@unisg.ch

David Durward

Kassel University, Germany

david.durward@uni-kassel.de

Ivo Blohm

University of St.Gallen, Switzerland

ivo.blohm@unisg.ch

Jan Marco Leimeister

University of St.Gallen, Switzerland

janmarco.leimeister@unisg.ch

Abstract

As software development cycles become shorter and shorter, while software complexity increases and IT budgets stagnate, many companies are looking for new ways of acquiring and sourcing knowledge outside their boundaries. One promising solution to aggregate know-how and manage large distributed teams in software development is crowdsourcing. This paper analyzes the existing body of knowledge regarding crowdsourcing in software development. As a result, we propose a fundamental framework with five dimensions to structure the existing insights of crowdsourcing in the context of software development and to derive a research agenda to guide further research.

Keywords: Crowdsourcing, Software, Development, Literature Review

1 Introduction

Faced with an increasingly dynamic environment, shorter product lifecycles, cost pressure, and an increasing complexity due to the rapid development of new software-based business models and a fragmented hardware market, companies are looking for new ways of acquiring and sourcing knowledge from outside the boundaries of their units, functions, or even outside their organization in order to develop software solutions (Jain 2010). On top of the continuous trend towards globalization and its focus on collaborative methods and infrastructure, it fosters the emergence of developing software in large distributed teams and communities (Boehm 2006; Stol and Fitzgerald 2014a). One solution to manage large distributed teams is crowdsourcing. With crowdsourcing, companies can reach out to the masses (Vukovic 2009) and open tasks to what Howe (2006) describes as "an undefined (…) network of people". The term itself derives from the concept of the outsourcing of a corporate, company-internal task to an independent mass of people, the crowd (Howe 2008).

IT industry leaders such as Fujitsu-Siemens (Füller et al. 2011), IBM (Bjelland and Wood 2008), or SAP (Blohm et al. 2011; Leimeister et al. 2009) already leveraged the "wisdom of the crowds" (Surowiecki 2005) for improving innovation management. Similarly, Lakhani et al. (2013) exhibit the tremendous potential of crowdsourcing in the domain of software development. They report on a programming contest in which about 75% of the submitted algorithms to solve an immunogenomic problem outperformed the industry standard while the total cost of the contest equaled 6000$. Extreme solutions were up to a thousand times faster than the industry standard. Software testing is another field of application in software development in which crowdsourcing is gaining importance. The World Quality Report (2014), the benchmark for software testing practices, indicates that more than half of the asked organizations either already employed crowdsourcing in their software testing process or planned to do so in 2014.

However, research on crowdsourcing is still in its inception. So far, crowdsourcing research has predominantly focused on (1) conceptualizing the phenomenon and comparing and designing, coding, testing, and documenting software. We intend to tackle this issue by reviewing existing crowdsourcing literature with a structured and systematic literature review following Webster & Watson (2002) and Vom Brocke et al. (2009). Based on this review, we propose a framework that summarizes existing research on crowdsourced software development. Following this research goal, our paper contributes to crowdsourcing literature by providing a basis for future theory development while elaborating various avenues for future research.

The remainder of this paper is structured as follows: Section two covers the literature review. Within this section, we first define the review scope and conceptualize the topic. Following that, we describe the literature search approach and introduce the literature framework. In section three, we present our findings in order to derive and discuss the research agenda which is presented in section four. Finally, we point out limitations and conclude the paper by summarizing the results of the literature review.

contrasting it to related phenomena such as collective intelligence (Malone et al. 2010), human computation, or open innovation (Gassmann et al. 2010), (2) classifying socio-technical crowdsourcing systems with taxonomies and categorizations to identify the basic characteristics (Geiger et al. 2011; Rouse 2010), and (3) applying crowdsourcing in different domains such as innovation development or marketing (Brabham 2008; Burger-Helmchen and Penin 2010; Kittur et al. 2008; Zhao and Zhu 2012). The thereby generated insights provide first references for the management and organization of crowdsourcing initiatives. Although there are already numerous research projects examining crowdsourced software development, e.g., Lakhani et al. (2013), Nag et al. (2012), and Liu et al. (2012) on the application of crowdsourced software development or Murray-Rust et al. (2014) and Wu et al. (2013a; 2013b) on system and process design, there are much less as well as no structured insights on research of crowdsourced software development. Lacking are theories and approaches to gain a deeper understanding and to systematically use crowdsourcing in Literature Review

This literature review is based on Vom Brocke et al.'s (2009) framework for reviewing scholarly literature and comprises five steps: (1) defining the review scope, (2) conceptualizing the topic, (3) searching for literature, (4) analyzing and synthesizing the literature, and (5) deriving a research agenda.

1.1 Definition of the Review Scope

The first step of a rigorous literature review is the definition of the review scope for which we follow the taxonomy of Cooper (1988). The paper focuses on research outcomes and the applications of crowdsourced software development (1). The goal of the literature review is to build an integrative (2) overview of the existing body of knowledge to present the state of the art (4) as it addresses specialized scholars (5). Table 1 depicts the literature review scope.

<table>
<tr><th colspan="2">Characteristics</th><th colspan="12">Categories</th></tr>
<tr><td>1</td><td>Focus</td><td colspan="3">Research Outcomes</td><td colspan="3">Research Methods</td><td colspan="3">Theories</td><td colspan="3">Applications</td></tr>
<tr><td>2</td><td>Goal</td><td colspan="4">Integration</td><td colspan="4">Criticism</td><td colspan="4">Central Issues</td></tr>
<tr><td>3</td><td>Organization</td><td colspan="4">Historical</td><td colspan="4">Conceptual</td><td colspan="4">Methodological</td></tr>
<tr><td>4</td><td>Perspective</td><td colspan="6">Neutral Representation</td><td colspan="6">Espousal of Position</td></tr>
<tr><td>5</td><td>Audience</td><td colspan="3">Specialized Scholars</td><td colspan="3">General Scholars</td><td colspan="3">Practitioners</td><td colspan="3">General Public</td></tr>
<tr><td>6</td><td>Coverage</td><td colspan="3">Exhaustive</td><td colspan="3">Exhaustive & Selective</td><td colspan="3">Representative</td><td colspan="3">Central/pivotal</td></tr>
</table>

Table 1: Definition of the Review Scope

1.2 Conceptualization of the Topic

A rigor literature review has to "provide a working definition of key variables" (Webster and Watson 2002). This work focuses on crowdsourcing and software development.

1.2.1 Crowdsourcing

Crowdsourcing describes a new form of outsourcing tasks, or more accurately, value creation activities and functions. The term itself is a neologism that combines crowd and outsourcing (Rouse 2010), introduced by Howe (2008), who defines crowdsourcing as "the act of taking a job traditionally per-formed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call". Whereas outsourcing describes the outplacement of specific corporate tasks to a designated third-party contractor or a certain institution, in crowdsourcing the tasks are allocated to an undefined mass of anonymous individuals, who are in turn rewarded for their effort of performing the tasks (Zogaj et al. 2014). In a crowdsourcing model, a firm or some type of institution first selects specific internal tasks it intends to crowdsource and subsequently broadcasts the underlying tasks online, i.e., via a crowdsourcing platform. In a second step, individuals (e.g., users registered on a crowdsourcing platform) self-select to work on the task solutions – either individually or in a collaborative manner – and subsequently submit the elaborated solutions via the crowdsourcing platform (Zogaj et al. 2014). The submissions are then assessed and – in case of successful completion – remunerated by the initiating organization. Hence, in a crowdsourcing model, at least two types of actors are engaged: the initiating organization that crowdsources specific tasks as well as the individuals from the crowd who perform these tasks. We denote the first entity as the crowdsourcer ["system owner" (Doan et al. 2011); "designated agent" (Howe 2006)]. The latter, namely the undefined contractors from the crowd, we label as crowdworkers since they perform the work (i.e., jobs or – more specifically – the tasks) that is outsourced by crowdsourcers. In most crowdsourcing initiatives, there is also a third type of agent: the crowdsourcing intermediary (also referred to as "crowdsourcing marketplace"; see e.g., Vukovic (2009) and Ipeirotis (2010). Crowdsourcing intermediaries mediate the process between the crowdsourcer and the crowdworkers by providing a platform for interaction between the parties. However, in some rare cases, the crowdsourcer establishes and hosts its own crowdsourcing platform such that an intermediary is not necessary.

1.2.2 Crowdsourced Software Development

In an early definition, Robillard (1999) describes software development as the processing of knowledge in a very focused way as well as a progressive crystallization of knowledge into a language that can be read and executed by a computer. This language creation is increasingly taking place in a steady, irreversible trend toward the globalization of business, in particular in software-intensive high-technology businesses. Hence, software has become an essential component of almost any value chain, and success in business increasingly depends on using software as a competitive weapon (Herbsleb and Moitra 2001). In the era of cloud computing, mobile computing, collaboration, and big data, software development and its requirements are significantly changing. Organizations as well as the users of software are calling for an improved ease of use, shorter development cycles, and a better integration by lower overhead operations (Huhns et al. 2013). This leads to more flexible and effective ways to build software solutions such as crowdsourcing software development. This approach uses the online crowd to outsource (sub-) tasks including requirements, design, coding, testing, evolution, and documentation. Crowdsourcing software development represents a paradigm shift from conventional industrial software development to a crowdsourcing-based peer-production software development and can be seen as next-generation outsourcing or offshoring (Huhns et al. 2013).

1.3 Literature Search

In order to identify relevant articles and to assure a rigorous, comprehensive, and traceable literature search, a systematic literature review was conducted (Vom Brocke et al. 2009). First, a journal search was executed, followed by a database search with keywords. Finally, a forward and backward search of citation indexes was conducted.

The journal search is the first step as major contributions are likely to be found in leading journals (Webster and Watson 2002) as well as in proceedings of highly ranked conferences (Rowley and Slack 2004). For the journal search, leading journals from Information Systems (IS) and Software Engineering (SE) were considered. For information systems, these included: Information Systems Research (ISR), MIS Quarterly (MISQ), Journal of Information Systems (JIS), and the Journal of Management Information Systems (JMIS). For software engineering, the highest ranked journals according to the ISI Web of Science were chosen, i.e., IEEE Transactions on Software Engineering, Communications of the ACM, IEEE Software, and IEEE Computer. The selection of relevant conferences included the International Conference on Information Systems (ICIS), the European Conference on Information Systems (ECIS), and the American Conference on Information Systems (AMCIS) as well as the Hawaii International Conference on System Sciences (HICCS). For Software Engineering, the International Conference on Software Engineering (ICSE), Foundations of Software Engineering (FSE), International Test Conference (ITC), and Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA) were considered. Furthermore, the following databases were queried: EBSCOhost, Web of Science, ProQuest, ScienceDirect, as well as IEEE Xplore database, since the topic is also at the interface to software engineering.

Core of a literature search is the keyword search. According to the above defined key variables, the keyword search was conducted in afore mentioned databases with the following search strings: (1) “crowdsourcing” AND “software development”, (2) “crowdsourcing” AND “software”, and (3) “crowdsourcing” AND “software engineering”, as well as (4) “crowd” AND “software”, (5) “crowd” AND “software development”, and (6) “crowd” AND “software engineering”. Additionally, the keyword search contained the following search strings in order to increase the coverage: (7) “crowdsourcing” AND “software testing” and (8) “crowdtesting”. The literature search closed with a forward and backward search (Levy and Ellis 2006). Table 2 depicts the detailed result auf the literature search.

Search String	**EBSCO Host(BSP)**		**Web of Science**		**ProQuest**		**Science Direct**		**IEEE Xplore**		**TOTAL**	
	Hits	Review-ed	Hits	Reviewed	Hits	Review-ed	Hits	Review-ed	Hits	Review-ed	Hits	Review-ed
“crowdsourcing” AND “software development”	5	1	28	3	20	3	5	0	29	11	87	18
“crowdsourcing” AND “software”	19	1	63	5	65	3	17	1	117	10	281	20
“crowdsourcing” AND “software engineering”	3	0	3	1	10	1	2	0	44	7	62	9
“crowd” AND “software"	21	0	183	3	42	2	25	2	431	5	702	12
“crowd” AND “software development”	7	1	29	1	38	4	3	2	79	8	156	16
“crowd” AND “software engineering”	2	0	7	0	28	1	1	0	130	6	168	7
“crowdsourcing” AND “software testing”	2	0	4	3	10	3	4	2	16	6	36	14
“crowdtesting”	0	0	1	1	1	0	0	0	1	1	3	2
TOTAL	59	3	318	17	214	17	57	7	847	54	1495	98

Table 2: Results of the Literature Search per Database

1.4 Literature Analysis and Synthesis

The literature review identified a total of 27 relevant papers. Considering the publication dates, it is no surprise that crowdsourcing in software development is at an early stage of scientific research, since crowdsourcing itself is still an emerging research topic. Only one paper was published before 2012 (Kazman and Chen 2009). More than 85% of all identified relevant papers were published in 2013 or later. Figure 1 depicts the publications per year. Another indication of the early stage of this field of research is that not a single paper was published in one of the major and leading journals. The articles were rather published in smaller and specialized journals or at conferences. Overall, more than two thirds of the relevant papers are from the field of software engineering.

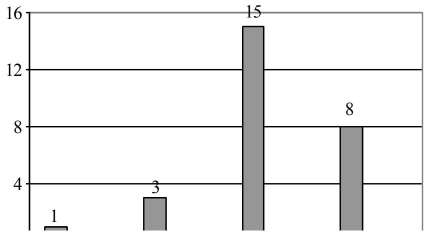

Figure 1: Publications per Year

In order to synthesize the literature, appropriate categories need to be developed. This paper tackles this issue by developing categories based on existing literature on crowdsourcing in general. Based on Zhao and Zhu's (2012) research roadmap, a key role-based perspective (Vukovic 2009; Zogaj et al. 2014), and applications of crowdsourcing in a software development context, the following categories were developed: (1) organization perspective, (2) intermediary perspective, (3) system perspective, (4) user perspective, and (5) application and evaluation.

(1) *Organization perspective*

In the archetypical crowdsourcing process (Vukovic 2009; Zogaj et al. 2014), organizations appear as the requester of a crowdsourcing task (crowdsourcer). This category sums up papers dealing with the organizational implementation, its according challenges, as well as the development of necessary capabilities to harness crowdsourcing in an enterprise environment.

(2) Intermediary perspective

The intermediary manages the crowdsourcing process and thereby its customers, crowd, and technology (Zogaj et al. 2014). This category sums up papers addressing process and design requirements, an according evaluation, as well as other managerial challenges the intermediary faces in crowdsourced software development tasks.

(3) System perspective

Crowdsourcing systems are socio-technical systems to enable and support the crowdsourcing process (Zhao and Zhu 2012). This category sums up papers dealing with the requirements or the design of crowdsourcing platforms for software development. Since software development tasks are way more complex than simple tasks that are frequently crowdsourced, it might take other design principles to develop a system tailored for software development tasks.

(4) User perspective

The participants of crowdsourcing initiatives (crowdworkers) are without a doubt an essential part and therefore need to be treated as a partner. By means of crowdsourcing, participants can expand their working experiences or even turn their hobbies into something beneficial (Zhao and Zhu 2012). This category sums up papers dealing with user motivation, payoff, and other user-centered aspects.

(5) Application and evaluation

The last category sums up papers which apply crowdsourcing in different contexts to evaluate its performance and/or highlight application possibilities for crowdsourcing in different software development contexts and stages.

2 Findings

Overall, it can be stated that existing research in the field of crowdsourcing software development mainly focuses on crowdsourcing systems and applications. Almost 60% of the investigated literature dealt with a particular IT system and its design (system perspective). About two fifths of the research dealt with the application of crowdsourcing in software development. Only one paper addresses the user perspective in crowdsourced software development. Table 3 depicts the detailed results.

Paper	Organization	Intermediary	System	User	Application
Amini et al. (2012)			x		x
Chen and Luo (2014)					x
Dolstra et al. (2013)			x		x
Hossfeld et al. (2014)			x		x
Hu and Wu (2014)			x	x	
Jayakanthan and Sundararajan (2013)					x
Kazman and Hong-Mei Chen (2009)			x		
Lakhani et al. (2013)					x
LaToza et al. (2013)			x		
Li et al. (2013)			x		
Liu et al. (2012)					x
Mäntylä and Itkonen (2013)					x
Mao et al. (2013)		x	x		
Murray-Rust et al. (2014)			x		
Musson et al. (2013)					x
Nag et al. (2012)			x		x
Pastore et al. (2013)			x		
Peng et al. (2014)		x			
Ponzanelli et al. (2013)			x		x
Stol and Fitzgerald (2014a)	x				
Stol and Fitzgerald (2014b)	x				
Stol and Fitzgerald (2014c)	x				
Tajedin and Nevo (2013)			x		
Tung and Tseng (2013)			x		
Wu et al. (2013a)			x		
Wu et al. (2013b)			x		
Zogaj et al. (2013)		x			x
TOTAL (n=27)	3 (11.1%)	3 (11.1%)	16 (59.3%)	1 (3.7%)	12 (44.4%)

Table 3: Literature Synthesis

(1) Organization perspective

So far, most notably Stol and Fitzgerald (2014a; 2014b; 2014c) examined crowdsourced software development from an organization's point of view. Their contribution to the understanding of crowdsourcing software development is twofold. First, they point out potential benefits and thus deliver a first explanation of why companies tend to use crowdsourcing in this area. The benefits combine traditional outsourcing benefits such as cost reduction, a faster time-to-market, and higher quality (Dibbern et al. 2004) with benefits of crowdsourcing such as creativity, increased openness, and diverse solutions (Afuah and Tucci 2012; Leimeister 2010). Second, they develop a framework of key concerns regarding the application of crowdsourcing for enterprises in software development. According key concerns are (1) task decomposition, (2) coordination and communication, (3) planning & scheduling, (4) quality assurance, (5) knowledge & IP, and (6) motivation & remuneration (Stol and Fitzgerald 2014a).

(2) Intermediary perspective

Zogaj et al. (2014) deliver a profound overview of the challenges of an intermediary from a managerial point of view. In their case study, they explicitly address the challenges in managing a crowd, the crowdsourcing process, as well as the crowdsourcing platform. Besides managing the process itself, the pivotal challenge for intermediaries constitutes building virtual teams and fostering collaboration among the crowdworkers (Peng et al. 2014). Furthermore, Mao et al. (2013) address the pricing of programming competitions, finding that the main antecedents of project pricing are whether the task is a component update or a new component, the size and the amount of illustrations in the specification document, as well as the overall size of the project and the posted reward amount.

(3) System perspective

Current literature mostly focuses on a system perspective. That means the development and derivation of specific development models, design principles for platforms or processes to enable crowdsourcing in diverse fields of application in software development. Therefore, the majorities of the papers have a technical perspective. Kazman & Chen (2009) and LaToza et al. (2013) propose a specific software development model tailored for crowdsourcing. Moreover, the human-machine interaction process is crucial for successful crowdsourcing campaigns. Murray-Rust et al. (2014) elaborate two collaboration models for community-based development of software and provide a conceptual model for combining process models with crowdsourced teams. Other research deals with success factors of crowdsourcing projects (Li et al. 2013; Tajedin and Nevo 2013) or process design for specific applications and purposes (Amini et al. 2012; Pastore et al. 2013; Tung and Tseng 2013). Furthermore, Wu et al. (2013a; 2013b) analyze software crowdsourcing processes by examining their key characteristics. They propose a novel evaluation framework for software crowdsourcing processes. Hossfeld et al. (2014) present key issues in the field of QoE-Testing (quality of experience) as they apply a QoE-Test and provide design guidelines for crowdtesting in this field of application.

(4) User perspective

As defined in the synthesis, this category clusters papers investigating the motivation and behavior of crowdworkers. Hu & Wu (2014) apply a game-theoretic approach to better understand the competitive behavior of crowdworkers in software development challenges.

(5) Application and evaluation

There are multiple examples for the application and evaluation addressing multiple parts of software development stages and functions. This research reaches from algorithm development (Lakhani et al. 2013), to embedded software for space robotics (Nag et al. 2012), or software testing (Mäntylä and Itkonen 2013). Further, Chen and Luo (2014) apply crowdsourced software testing in an educational context. As part of their studies, students had to test several web and mobile applications. Liu et al. (2012) compare traditional laboratory-based usability testing with crowdsourced usability testing, indicating that the acquisition of testers through crowdsourcing is much easier at significantly lower costs . Contrariwise, the received feedback per participant was less informative. Crowdsourcing also seems to be a promising approach to test graphical user interfaces or to evaluate mobile applications (Amini et al. 2012; Dolstra et al. 2013). In the domain of documenting software code, Ponzanelli et al. (2013) research the case of "Stack Overflow", the world's largest language-independent collaboratively edited question and answer site for programmers. They propose a new interaction interface for increasing the productivity of software documentation. The power of crowdsourcing has also been used to monitor software performance. One major advantage is the "real world setting" in which different network environments and bandwidths are accessible, which are not to be covered in laboratory tests (Musson et al. 2013). Jayakanthan & Sundararajan (2012) introduce a prototype for a corporate crowdsourcing solution at TCS, one of the largest IT consulting and software development companies worldwide. The crowdsourcing system will unify three modes of crowdsourcing as the crowdsourcer can select whether to choose a single expert among crowdworkers performing the task (e.g., when special knowledge is required or the task is critical), recruit a group of crowdworkers (e.g., for software testing), or create a competition to choose the best solution among the submissions (e.g., for coding).

3 Discussion

3.1 Research Agenda

Although various scholars have examined crowdsourced software development projects, our literature review reveals that research in this regard is still at an early stage. Existing literature most notably focuses on the design of crowdsourcing platforms from a *system perspective* providing only little generalization and approaches to gain a broad perspective.

At first sight, there is obviously more research needed on the *user perspective* since we found only one paper which addresses this stream within the literature. But it is important to note that a wide body of research regarding the incentives and motivation of crowdworkers (Muhdi and Boutellier 2011; Pilz and Gewald 2013), qualification, as well as the impact on work conditions and labor rights (Brabham 2012) already exists in broader crowdsourcing literature. However, most of this research deals with simplistic “micro tasks” or highly creative tasks such as idea generation. Thus, future research has to validate that the findings are also applicable to software development – a more complicated knowledge task. Crowdsourced software development has been *applied and evaluated* in a number of studies. So far, there is no in-depth knowledge on the basic conditions of crowdsourcing projects to leverage this potential. Software development is a very complex process with diverse stages and tasks with very different requirements, complexity, modularity, and structures – things which have all been found to determine the effectivity of crowdsourcing (Afuah and Tucci 2012; Stol and Fitzgerald 2014a). Future research should address this aspect by examining which tasks can be crowdsourced and how crowdsourcing projects should be structured in terms of task decomposition with respect to the knowledge intensity and a high degree of complexity in software development.

A third possible stream is based on these insights and addresses the *organization perspective.* The literature review has shown first activity regarding this topic with Stol & Fitzgerald (2014a) identifying six “key concerns” in crowdsourcing software development. To eliminate these concerns, it is not sufficient to solely understand at which stage in the development process they can crowdsource tasks and how to structure this work. Just like outsourcing, crowdsourcing is a whole new form of organizing work and therefore requires a different process model, as well as governance structures and control mechanisms in software development projects. To systematically enable organizations to conduct crowdsourcing projects within their existing process and framework, we propose to develop a reference model for crowdsourcing projects which addresses the key concerns and guides organizations through the crowdsourcing process.

From an *intermediary perspective*, Zogaj et al. (2014) discuss the challenges an intermediary faces in the crowdsourcing process. As these intermediaries appear as interface between the crowdsourcer and the crowdworkers, two research fields unfold. The first overlaps with the user perspective as it is crucial for the intermediary to promote a motivated and active crowd. Further research should investigate the question of how to motivate the users, especially for less entertaining tasks such as software documentation. On the other side, the intermediary is a vital part in the crowdsourcing process from an organization’s point of view. Further research should investigate the intermediary’s role and support in crowdsourcing software development. In this regard, investigating crowdsourcing intermediaries as two-sided markets may be of particular interest.

3.2 Limitations

Reflecting the paper, two limitations are worth mentioning. First, the results only rely on scientific literature and thus lack insights from practice. Second, this research only focuses on core crowdsourcing literature as the search strings only contained “crowd” or “crowdsourcing”. However, there are other streams of research that might be suitable to address some of the key issues. For instance, the concept of open source software development is a more mature research field and overlaps with the concept of crowdsourcing. Further research should target this research stream in order to integrate and enable a deeper understanding of how collaborative work in software development can be organized and processed.

4 Conclusion

In summary, the research on crowdsourcing in software development is still limited, despite its potential and gaining importance in organizations. In this paper, we reviewed the existing body of literature regarding crowdsourced software development. In so doing, the contribution of this paper is twofold. First we provide an initial framework summarizing the key aspects of crowdsourcing and thus contribute to an enhanced development of a theoretical understanding of crowdsourcing. Second, our literature review points out gaps in the literature that could be addressed in future research.

References

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

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

Bjelland, O.M., and Wood, R.C. 2008. "An inside View of Ibm's' Innovation Jam'," *MIT Sloan management review* (50:1), pp. 32-40.

Blohm, I., Bretschneider, U., Leimeister, J.M., and Krcmar, H. 2011. "Does Collaboration among Participants Lead to Better Ideas in It-Based Idea Competitions? An Empirical Investigation," *International Journal of Networking and Virtual Organisations* (9:2), pp. 106-122.

Boehm, B. 2006. "A View of 20th and 21st Century Software Engineering," *Proceedings of the 28th international conference on Software engineering*: 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.

Brabham, D.C. 2012. "The Myth of Amateur Crowds: A Critical Discourse Analysis of Crowdsourcing Coverage," *Information, Communication & Society* (15:3), pp. 394-410.

Buenen, M., and Makarand, T. 2014. "World Quality Report 2014-2015."

Burger-Helmchen, T., and Penin, J. 2010. "The Limits of Crowdsourcing Inventive Activities,").

Chen, Z., and Luo, B. 2014. "Quasi-Crowdsourcing Testing for Educational Projects," *Companion Proceedings of the 36th International Conference on Software Engineering*: ACM, pp. 272-275.

Cooper, H.M. 1988. "Organizing Knowledge Syntheses: A Taxonomy of Literature Reviews," *Knowledge in Society* (1:1), pp. 104-126.

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.

Doan, A., Ramakrishnan, R., and Halevy, A.Y. 2011. "Crowdsourcing Systems on the World-Wide Web," *Communications of the ACM* (54:4), pp. 86-96.

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

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.

Gassmann, O., Enkel, E., and Chesbrough, H. 2010. "The Future of Open Innovation," *R&D Management* (40:3), pp. 213-221.

Geiger, D., Seedorf, S., Schulze, T., Nickerson, R.C., and Schader, M. 2011. "Managing the Crowd: Towards a Taxonomy of Crowdsourcing Processes,").

Herbsleb, J.D., and Moitra, D. 2001. "Global Software Development," *Software, IEEE* (18:2), pp. 16-20.

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," *Multimedia, IEEE Transactions on* (16:2), pp. 541-558.

Howe, J. 2006. "The Rise of Crowdsourcing," *Wired magazine* (14:6), pp. 1-4.

Howe, J. 2008. *Crowdsourcing: How the Power of the Crowd Is Driving the Future of Business*. Random House.

Hu, Z., and Wu, W. 2014. "A Game Theoretic Model of Software Crowdsourcing," *Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*: IEEE, pp. 446-453.

Huhns, M.N., Li, W., and Tsai, W.-T. 2013. "Cloud-Based Software Crowdsourcing," Citeseer.

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

Jain, R. 2010. "Investigation of Governance Mechanisms for Crowdsourcing Initiatives,").

Jayakanthan, R., and Sundararajan, D. 2012. "Enterprise Crowdsourcing Solution for Software Development in an Outsourcing Organization," in *Current Trends in Web Engineering*. Springer, pp. 177-180.

Kazman, R., and Chen, H.-M. 2009. "The Metropolis Model a New Logic for Development of Crowdsourced Systems," *Communications of the ACM* (52:7), pp. 76-84.

Kittur, A., Chi, E.H., and Suh, B. 2008. "Crowdsourcing User Studies with Mechanical Turk," *CHI '08 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (453-456).

Lakhani, K.R., Boudreau, K.J., Loh, P.-R., Backstrom, L., Baldwin, C., Lonstein, E., Lydon, M., MacCormack, A., Arnaout, R.A., and Guinan, E.C. 2013. "Prize-Based Contests Can Provide Solutions to Computational Biology Problems," *Nature biotechnology* (31:2), pp. 108-111.

LaToza, T.D., Towne, W.B., Van Der Hoek, A., and Herbsleb, J.D. 2013. "Crowd Development," *Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on*: IEEE, pp. 85-88.

Leimeister, J.M. 2010. "Collective Intelligence," *Business & Information Systems Engineering* (2:4), pp. 245-248.

Leimeister, J.M., Huber, M., Bretschneider, U., and Krcmar, H. 2009. "Leveraging Crowdsourcing: Activation-Supporting Components for It-Based Ideas Competition," *Journal of management information systems* (26:1), pp. 197-224.

Levy, Y., and Ellis, T.J. 2006. "A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research," *Informing Science: International Journal of an Emerging Transdiscipline* (9:1), pp. 181-212.

Li, K., Xiao, J., Wang, Y., and Wang, Q. 2013. "Analysis of the Key Factors for Software Quality in Crowdsourcing Development: An Empirical Study on Topcoder. Com," *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual*: IEEE, pp. 812-817.

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.

Malone, T.W., Laubacher, R., and Dellarocas, C. 2010. "The Collective Intelligence Genome," *IEEE Engineering Management Review* (38:3), p. 38.

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.

Mao, K., Yang, Y., Li, M., and Harman, M. 2013. "Pricing Crowdsourcing-Based Software Development Tasks," *Proceedings of the 2013 International Conference on Software Engineering*: IEEE Press, pp. 1205-1208.

Muhdi, L., and Boutellier, R. 2011. "Motivational Factors Affecting Participation and Contribution of Members in Two Different Swiss Innovation Communities," *International Journal of Innovation Management* (15:03), pp. 543-562.

Murray-Rust, D., Scekic, O., Truong, H.-L., Robertson, D., and Dustdar, S. 2014. "A Collaboration Model for Community-Based Software Development with Social Machines," *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*: IEEE, pp. 84-93.

Musson, R., Richards, J., Fisher, D., Bird, C., Bussone, B., and Ganguly, S. 2013. "Leveraging the Crowd: How 48,000 Users Helped Improve Lync Performance," *Software, IEEE* (30:4), pp. 38-45.

Nag, S., Heffan, I., Saenz-Otero, A., and Lydon, M. 2012. "Spheres Zero Robotics Software Development: Lessons on Crowdsourcing and Collaborative Competition," *Aerospace Conference, 2012 IEEE*: IEEE, pp. 1-17.

Pastore, F., Mariani, L., and Fraser, G. 2013. "Crowdoracles: Can the Crowd Solve the Oracle Problem," *International Conference on Software Testing, Verification and Validation (ICST)*.

Peng, X., Babar, M.A., and Ebert, C. 2014. "Collaborative Software Development Platforms for Crowdsourcing," *IEEE software* (31:2), pp. 30-36.

Pilz, D., and Gewald, H. 2013. "Does Money Matter? Motivational Factors for Participation in Paid-and Non-Profit-Crowdsourcing Communities,").

Ponzanelli, L., Bacchelli, A., and Lanza, M. 2013. "Leveraging Crowd Knowledge for Software Comprehension and Development," *Proceedings of the 2013 17th European Conference on Software Maintenance and Reengineering*: IEEE Computer Society, pp. 57-66.

Robillard, P.N. 1999. "The Role of Knowledge in Software Development," *Communications of the ACM* (42:1), pp. 87-92.

Rouse, A.C. 2010. "A Preliminary Taxonomy of Crowdsourcing," *Australian Conference on Information Systems (ACIS)*, 1-3 Dec 2010, Brisbane.

Rowley, J., and Slack, F. 2004. "Conducting a Literature Review," *Management Research News* (27:6), pp. 31-39.

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

Stol, K.-J., and Fitzgerald, B. 2014b. "Researching Crowdsourcing Software Development: Perspectives and Concerns," *Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering*: ACM, pp. 7-10.

Stol, K.-J., and Fitzgerald, B. 2014c. "Research Protocol for a Case Study of Crowdsourcing Software Development."

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

Tajedin, H., and Nevo, D. 2013. "Determinants of Success in Crowdsourcing Software Development," *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.

Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., and Cleven, A. 2009. "Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process," *ECIS*, pp. 2206-2217.

Vukovic, M. 2009. "Crowdsourcing for Enterprises," *Services-I, 2009 World Conference on*: IEEE, pp. 686-692.

Webster, J., and Watson, R.T. 2002. "Analyzing the Past to Prepare for the Future: Writing a Literature Review," *Management Information Systems Quarterly* (26:2), p. 3.

Wu, W., Tsai, W.-T., and Li, W. 2013a. "An Evaluation Framework for Software Crowdsourcing," *Frontiers of Computer Science* (7:5), pp. 694-709.

Wu, W., Tsai, W.T., and Li, W. 2013b. "Creative Software Crowdsourcing: From Components and Algorithm Development to Project Concept Formations," *International Journal of Creative Computing* (1:1), pp. 57-91.

Zhao, Y., and Zhu, Q. 2012. "Evaluation on Crowdsourcing Research: Current Status and Future Direction," *Information Systems Frontiers*), pp. 1-18.

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.