Breaking Collaboration Barriers through Communication Practices in Software Crowdsourcing

Leticia Machado, Josiane Kroll, Sabrina Marczak, Rafael Prikladnicki
MUNDDOS Research Group - Computer Science School
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil
{leticia.machado.001, josiane.kroll}@acad.pucrs.br, {sabrina.marczak, rafael.prikladnicki}@pucrs.br

***Abstract*—Software Crowdsourcing (SW CS) is an emerging area in Software Engineering (SE). It has a dynamic nature in which problems are opened up to crowds of people through an open call to solve problems with the incentive of prizes for the best solutions. Although SW CS has been increasingly adopted in the software industry, many open issues are still to be elucidated. For instance, collaboration aspects are one of those issues. They impact communication and coordination practices. This study aims to identify barriers to collaboration in SW CS and communication practices that might help overcome these barriers. We identified the barriers and practices from a qualitative data analysis study considering an available literature review report. Our study contributes to the SE area by providing insights for researchers, tool designers, and managers that might help them to better understand collaboration issues and proposed solutions to solve them.**

Keywords – Software Engineering; Crowdsourcing; Collaboration; Communication; Practice; Issue; Barrier.

I. INTRODUCTION

The Crowdsourcing (CS) phenomenon emerged from new collaboration technologies such as social media and Web 2.0 [1, 2]. This term is coined by Howe and Robison [2] and it is part of concepts such as mass collaboration, open collaboration, and collective intelligence. This new form of work in which 'crowds' of people can collaborate and complete a software task (e.g., coding, testing) is an alternative to software organizations. While the overall impact has been mundane until now, SW CS has the potential to and will impose disruptive changes in how software will be developed in the next years [8, 9].

SW CS is the engagement of a global pool of online workers who can be tapped on-demand in order to contribute to various types of software development tasks [12, 13, 14]. The CS process is mediated by platforms that connect requesters with online workers. The platform allows the requester to spread tasks to workers who volunteer to solve them motivated by compensation (e.g. monetary, prestige).

In SW CS, the platform has a relevant importance providing directions for the management and the coordination of processes and people in both technical and business levels [11, 13]. The platform also allows requesters to find out talents beyond their boundaries and take advantages of cost, time, quality, and expertise [8, 12, 13].

A recent study by Ågerfalk et al. [10] reports challenges in SW CS. Communication and collaboration among the crowd members are cited as open issues that lead to repeated occurrences of poor management practices in SW CS [11]. By building collaboration into a model that is based on competition, communication, coordination, and cooperation related to problems are likely to emerge [6, 15].

In this study, we look back to literature to identify barriers to collaboration and communication practices to solve them. We investigate the current state of collaboration and the underlying difficulties as well as the communication practices used to support them in SW CS platforms. Our results can potentially advance the theoretical foundation into the latent characteristics that affect collaborative relations in SW CS. In our qualitative analysis of literature, we found 36 collaboration barriers and 30 communication practices. Practices were associated to barriers in order to investigate the relationship between them.

II. RESEARCH METHOD

We conducted a qualitative mapping analysis in an already consolidated literature review report about SW CS to identify empirically-based evidences of barriers to collaboration in such context and communication practices that might be of help to reduce them. Our goal here was first to extract a list of barriers to collaboration followed by a set of communication practices and then propose an association between the lists.

For this study, we reviewed 241 papers. A data repository built by Mao et al. [20] was adopted to identify relevant papers. The database includes papers from diverse categories (e.g., peer-reviewed conference papers, journal articles, technical reports, thesis) published between January 2006 and May 2015. This repository is publicly available online[1]. We extracted 229 papers from this repository that fit our search criteria, which were: collaboration challenges or issues, communication practices, and empirical work. We also updated the dataset by including 10 new papers published from May 2015 to January 2016, which were not listed on the Mao et al. repository. We manually searched well-known databases such as *IEEEXplore Digital Library,*

[1] http://crowddev.kemao.uk/cse_repository

2329-6313/16 $31.00 © 2016 IEEE
DOI 10.1109/ICGSE.2016.32

ACM Digital Library, Springer Link Online Library, Elsevier ScienceDirect, and *Google Scholar.*

For each selected paper, one of the authors read the entire paper and extracted any barrier to collaboration or any communication practice cited in the respective study. We used a spreadsheet to catalogue the extracted data. Once we finished data extraction, we peer reviewed the results by selecting a sample and having one of the authors redoing the data analysis. Discrepancies were discussed between the 2 authors and resolved with the help of a third one. Next, again, one of the authors proposed the mapping between the collaboration barriers and the communication practices based on the papers selected. The mapping was then discussed among 3 of the authors and critically reviewed in several review sessions until it was considered stable.

TABLE I. COLLABORATION BARRIES

ID	Barriers (B)	Reference
B1	Informal communication	[13,21,22,24,26,47]
B2	Psychological issues of CS	[26, 42, 35, 6, 11]
B3	Interaction between parties	[25, 26, 18, 27]
B4	Software process methodologies	[13, 50, 51, 52]
B5	Keeping participants motivation	[29, 30, 11]
B6	Real-time collaboration	[31, 15, 32]
B7	Plattform to support social media	[43, 31, 32]
B8	Micro-task decomposition	[35, 13, 25]
B9	Scarce context for the crowd	[35, 25, 14]
B10	Collaboration among crowd members	[11, 29, 30]
B11	Different languages	[12, 23, 47]
B12	Reputation management in competitions	[26, 29]
B13	Technical and privacy issues	[33, 13]
B14	Large scale collaboration with distributed members	[11, 34]
B15	Orchestration of virtual teams	[11, 30]
B16	Reusable software components	[35, 36]
B17	Internal collaboration between the platform and the requester	[13, 25]
B18	Task allocation	[37, 13]
B19	Management in asynchronous communication environments	[19, 53]
B20	Software development infrastructure	[38]
B21	Management of shared version control systems and issue trackers	[11]
B22	Information management	[39]
B23	Self- management and real time crowd collaboration environment	[40]
B24	Collaboration between anonymous participants in the crowd	[41]
B25	Competition management	[42]
B26	Programming metaphors and infrastructures to support the design, and execution of human computation	[44]
B27	Commitment between involved parties	[45]
B28	Management a large amount of questions from the crowd during task execution period	[65]
B29	Single point of contact	[13]
B30	Documentation	[13]
B31	Reliability of remote participants and trust among crowd participants	[48]
B32	Reduced larger global project view	[35]
B33	Definition of the crowd's role	[46]
B34	Teams heterogeneity	[19]
B35	Crowd worker latency	[44]
B36	Collaboration between workers and employers outside the platform	[40]

III. RESULTS

Our analysis of the data revealed 36 barriers to collaboration and 30 communication practices. We present our findings in the next sections.

A. Collaboration Barriers

We extracted the barriers from the selected papers and ordered them based on the number of papers they are cited. Table I summarizes the identified list.

B. Communication Practices

We identified 30 communication practices adopted in SW CS. Similarly to the collaboration barriers, practices were also ordered by the number of papers they are cited as presented on Table II.

TABLE II. COMMUNICATION PRACTICES

ID	Practices (P)	Reference
P1	Asyncronous communication for knowledge sharing	[11,13,26,30,50, 27, 28,18,16,53, 54]
P2	Integration of the social network to known participants	[46, 18, 31, 53, 55, 6, 40, 31]
P3	Increasing communication between requester and crowd	[56, 35, 57, 40, 21, 52, 11]
P4	Group or global chat to provide tasks' details	[50,40,58,54,45, 59]
P5	Task coordinator to manage and answer crowd questions	[13, 25, 26, 30, 11]
P6	Feedback channel should be robust	[40, 62, 33, 25]
P7	Discussion boards and wikis	[50, 36, 58, 27]
P8	Collaborative software development infrastructure	[16, 27]
P9	Fine-grained explicit task	[59, 33]
P10	Process documentation	[27, 15]
P11	'*In momento'* approach	[59, 33]
P12	Communication between crowd to crowd	[11]
P13	Information assurance	[26]
P14	Community-driven and decentralized to indirectly freely	[39]
P15	Communication tools to facilitate trainings	[33]
P16	Mobile apps	[18]
P17	Continuous communication during task execution period	[21]
P18	Small groups of users for code sharing outside of the group	[60]
P19	Communicate by email new available tasks and their feedback after a particular task be done	[11]
P20	Task dashboard for the crowd	[11]
P21	Platforms should provide communication mechanisms to share artifacts	[11]
P22	Transparency through activity feed/timeline	[59]
P23	Iterative workflow to improve the design solutions from the crowd	[61]
P24	Programming tasks by types of context	[64]
P25	Free-text responses in HITs (Human Intelligence Tasks)	[63]
P26	Collaboration tools	[36]
P27	Annotations to share and track source code	[52]
P28	Simply inspecting a code location	[61]
P29	Simply understanding of the rationale underlying the decisions	[61]
P30	Crowd and requesters should know clearly the role and relationship of each CS element	[22]

IV. DISCUSSION

Our analysis focuses on their frequencies and on the relationship between the two variables: collaboration issues and communication practices. From the frequencies analysis it is possible to see which barriers have been emphasized in past research and thus to identify gaps and possibilities for future research. On the other hand, the relationship mapping analysis allows us to identify how communication practices help overcome collaboration barriers in SW CS.

We found that the greatest number of papers (6 papers) reports informal communication as the main challenge for collaboration in SW CS. It is possible to infer that the *B1: Informal communication* barrier is more frequently reported given that CS platforms tend to restrict the exchange of information between requesters and crowd [11]. Thus, it can impact in the frequency in which interactions occur between platform, requester, and crowd as corroborated by *B3: Interaction between parties*.

The second most cited barrier *B2: Psychological issues of CS* is reported in 5 papers. Given that SW CS development operates on the basis of competitions [13], these can impose a restriction to crowd members collaborate with one another. However, competitions can increase collaboration with other communities such private sector, non-profit, and academic [27]. This 'positive' side of it has to be further investigated. Same as B3, *B4: Software process methodologies* is also reported in 4 papers. We have observed that literature describe the adoption of traditional software processes methodologies for SW CS [13, 25]. However, traditional software processes do not fully meet collaboration in SW CS. Besides that, for the requester (company) side, it is not trivial to match different methodologies during the SW CS initiatives with internal software development [46].

In regards to the communication practices, 6 of them are the most reported in the papers (P1 to P7). Other 24 practices are reported by only two or one study each. The greatest number of papers recommends *P1: Asynchronous communication for knowledge sharing* (11 papers). Communication is needed between the developers and the customer [13, 25]. In contrast, Boudreau et al. [19] discuss topics hindered in asynchronous communication environments. In an asynchronous discussion, typically many topics are active at the same time, long time lapses between communication events can lead to discontinuous and seemingly disjointed discussions. Other six practices (P2 to P7) aim to increase communication between parties. These practices recommend the adoption of social network, global chat, the election of a task coordinator, and strong feedback to the participants.

The initial analysis of our findings shows the relationship between collaboration and communication. In real-world collaborations, interaction is the norm rather than the exception. There are many advantages to interacting groups, such as the ability to communicate. In order to show how collaboration and communication are related, we associated communication practices that help overcome collaboration difficulties in SW CS (see Table 3).

TABLE III. ASSOCIATION MAPPING

ID	Barriers	Mitigation Practices
B1	Informal communication	[1, 2, 3, 4, 12, 20, 22]
B2	Psychological issues of CS	[1, 2, 3, 4, 6, 12, 11,14, 18, 28]
B3	Interaction between parties	[1,2,3,4,6,10,12,15,17]
B4	Software process methodologies	[6, 7, 12, 14, 17, 28]
B5	Keeping participants motivation	[3, 17]
B6	Real-time collaboration	[4, 6,7,16,17,21,24,29]
B7	Plattform to support social media	[2, 16, 11]
B8	Micro-task decomposition	[3, 5, 9, 10]
B9	Scarce context for the crowd	[1, 4, 5, 10, 13]
B10	Collaboration among crowd members	[1, 4, 6, 10, 12, 21, 22, 23, 25]
B11	Different languages	[15]
B12	Reputation management in competitions	[13, 18, 24, 28,29]
B13	Technical and privacy issues	[18]
B14	Large scale collaboration with distributed members	[5,7,8,6,17,21,22,25]
B15	Orchestration of virtual teams	[1,14]
B16	Reusable software components	[11, 21, 26]
B17	Internal collaboration between the platform and the requester	[5]
B18	Task allocation	[9, 14, 17, 20, 30]
B19	Management in asynchronous communication environments	[3, 5, 6, 17, 19]
B20	Software development infrastructure	[8, 11, 26]
B21	Management of shared version control systems and issue trackers	[11, 21, 25, 26]
B22	Information management	[1, 3, 25]
B23	Self- management and real time crowd collaboration environment	[1, 2, 14]
B24	Collaboration between anonymous participants in the crowd	[1, 2, 3]
B25	Competition management	[8, 21, 24, 29]
B26	Programming metaphors and infrastructures to support the design, and execution of human computation	[8, 23, 27]
B27	Commitment between involved parties	[2, 7, 12, 21]
B28	Management a large amount of questions from the crowd during task execution period	[3, 5, 17, 18]
B29	Single point of contact	[3, 5, 12, 15, 17,11]
B30	Documentation	[9, 13, 21]
B31	Reliability of remote participants and trust among crowd participants	[1, 2, 3, 17]
B32	Reduced larger global project view	[1, 4, 5, 13]
B33	Definition of the crowd's role	[28]
B34	Teams heterogeneity	[2, 17]
B35	Crowd worker latency	[17]
B36	Collaboration between workers and employers outside the provided system	[2,19]

V. LIMITATIONS

Our study has some limitations. First, we analyzed data from a data repository that was not built for us. This data repository presents a list of papers related to the topic investigated in this study. We could only remove those that we consider did not fit our selection criteria. To reduce this limitation, we conducted a pilot search to compare our search results with the data repository list of papers. Since we did not find significant discrepancy in our results when compared to the data repository (aside from the 10 added papers), we assume it covers the main publications in the research field.

Second, the findings of this study may have also been affected as the data extraction and data classification processes were based on human understanding of the data, which is to a certain extent subjective, such as our own interpretation of what a paragraph states. To reduce this limitation, we preliminarily agreed upon the data analysis process and extraction criteria and later had additional researchers (co-authors) reviewing a sample of the findings as well as we hold a few rounds of consensus meetings until data was considered stable and final.

VI. Conclusion and Future Work

SW CS demonstrates a fundamental paradigm shift in how software will be developing in the future. Consequently, this raises a number of issues as well. SW CS taps global inputs to work, but it also increases complexity to decide which development tasks are more suitable to CS, cross-task coordination and virtual team organization, setting target audience, integrating crowd's deliverables, and ensuring quality of the software produced by the crowd.

In general, we have characterized collaboration barriers and communication practices in a particular domain (CW) and we believe that the SW CS platforms can benefit from our research to improve their software requirements.

Future research will involve gathering empirical evidence to assess our proposed mapping between collaboration barriers and communication practices to solve/reduce them. Also, to validate with a broader audience whether both lists are exhaustive. We will also focus on searching for commonalities and differences among barriers faced in different domains in order to develop models and theories about joining processes in SW CS communities. Researchers can benefit from this study by using the results to conceive strategies for collaboration support. By including the context of the SW CS and related research literature, we provide a starting point to conceive such support.

Acknowledgment

The authors acknowledge financial support from CNPq (projects 312127/2015-4 and 406692/2013-0), FAPERGS (project 2062-2551/13-7), and the PDTI Program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91).

References

[1] J. Howe, "Crowdsourcing: A definition," http:// crowdsourcing: typepad:com/cs/2006/06/crowdsourcinga:html, June 2006.

[2] ——, "The rise of crowdsourcing," Wired magazine, vol. 14, no. 6, pp. 1–4, 2006.

[3] S.A. Barnes, Sally, A. Green, and M. Hoyos, "Crowdsourcing and work: Individual factors and circumstances influencing employability," New Technology, Work and Employment, vol. 30, no. 1, pp. 16-31, 2015.

[4] E.E-Arolas and F.G-L-de-Guevara, "Towards an integrated crowdsourcing definition," Journal of Information Science, vol. 38, no. 2, pp. 189–200, 2012.

[5] D. C. Brabham, "Crowdsourcing as a model for problem solving an introduction and cases," Convergence: The international journal of research into new media technologies, vol. 14, no. 1, pp. 75–90, 2008.

[6] I. Boughzala, T. De Vreede, C. Nguyen, and G.J. De Vreede, "Towards a maturity model for the assessment of ideation in crowdsourcing projects," in: Proc. Hawaii International Conference in System Sciences, IEEE, Waikoloa, USA, 2014, p. 483-490.

[7] W. Wu, W.-T. Tsai, and W. Li, "An evaluation framework for software crowdsourcing," Frontiers of Computer Science, vol. 7, no. 5, pp. 694-709, 2013.

[8] T.D. LaToza and A. van der Hoek, "Crowdsourcing in software engineering: Models, motivations, and challenges," IEEE Software, vol. 33, no. 1, pp. 74-80, 2016.

[9] E. Kaganer, E. Carmel, R. Hirschheim, and Olsen, "Managing the human cloud," MIT Sloan Management Review, 2013.

[10] P.J. Agerfalk, B. Fitzgerald, and K. Stol, "Software outsourcing in the age of open: Leveraging the unknown workforce," Springer, pp. 45-60, 2015.

[11] X. Peng, M. Ali Babar, and C. Ebert, "Collaborative Software Development Platforms for Crowdsourcing," IEEE Software, vol. 31, no. 2, pp. 30–36, 2014.

[12] R. Prikladnicki, L. Machado, E. Carmel, and C. de Souza, "Brazil software crowdsourcing: a first step in a multi-year study," in: Proc. International Workshop on CrowdSourcing in Software Engineering. IEEE, Hyderabad, India, 2014.

[13] K.J. Stol and B. Fitzgerald, "Two's company, three's a crowd: A case study of crowdsourcing software development," in: Proc. Int'l Conf on Software Eng, IEEE, Hyderabad, India, 2014, p. 187–198.

[14] H. Tajedin and D. Nevo, "Determinants of success in crowdsourcing software development," in Annual Conference on Computers and People Research. ACM, Cincinnati, USA, 2013, p. 173.

[15] G.D. Saxton, O. Onook, and K. Rajiv, "Rules of crowdsourcing: Models, issues, and systems of control," Information Systems Management, vol. 30, no. 1, pp. 2-20, 2013.

[16] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng, "The impact of social media on software engineering practices and tools," in FSE/SDP Workshop on Future of Software Engineering Research, ACM, Santa Fe, USA, 2010, p. 359–364.

[17] D. L. Olson and K. Rosacker, "Crowdsourcing and open source software participation," Service Business, vol. 7, no. 4, pp. 499–511, 2012.

[18] A. Begel, J. Bosch, and M.-A. Storey, "Social networking meets software development: Perspectives from github, msdn, stack exchange, and topcoder," IEEE Sw, vol. 30, no. 1, pp. 52-66, 2013.

[19] K. Boudreau, P. Gaule, K. Lakhani, C. Riedl, and A. Woolley, "From crowds to collaborators: Initiating effort & catalyzing interactions among online creative workers", Harward Business School, Working Paper No. 14-060, 2014.

[20] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," Research Note No. 15/01 University College London, London, UK, 2015.

[21] M. Nayebi and G. Ruhe, "An open innovation approach in support of product release decisions," in: Int'l Workshop on Cooperative and Human Aspects of Software Eng, IEEE, Hyderabad, India, 2014.

[22] M. Saengkhattiya, M. Sevandersson, and U.Vallejo, "Quality in crowdsourcing: How software quality is ensured in software crowdsourcing," Master Thesis No. 15 ECTS, Lund University, 2012.

[23] T. W. Schiller and M.D. Ernst, "Reducing the barriers to writing verified specifications," in: Proc. SIGPLAN Notices – OOPSLA, ACM, Tucson, USA, 2012, p. 95-112.

[24] H. Tajedin and D. Nevo, "Value-adding intermediaries in software crowdsourcing," in: Proc. Hawaii International Conference on System Sciences, IEEE, Waikoloa, USA, 2014, p. 1396-1405.

[25] B. Fitzgerald and K.-J. Stol, "The dos and don'ts of crowdsourcing software development," in: SOFSEM: Theory and Practice of Computer Science. Lecture Notes in Computer Science, vol. 8939, 2015, pp. 58–64.

[26] K. Lakhani, A. D. Garvin, and E.Lonstein, "Topcoder (a): Developing software through crowdsourcing," Harvard Business School General Management Unit Case, 610-032, Jan. 2010.

[27] S. Nag, I. Heffan, A. Saenz-Otero, and M. Lydon, "SPHERES zZero Robotics software development: Lessons on crowdsourcing and

collaborative competition," in: Proc. Aerospace Conference, IEEE, Big Sky Resort , USA, 2012, p. 1–17.

[28] S.Nag, J.G. Katz, and A. S.Otero, "Collaborative gaming and competition using SPHERES Zero Robotics," CS-STEM Education Acta astronautica, vol. 83, pp. 145-174, 2013.

[29] M. Hosseini, A. Shahri, K. Phalp, and . Taylor, "Configuring crowdsourcing for requirements elicitation," in: Proc. International Conference on Research Challenges in Information Science, IEEE, Atenees, Greece, 2015, p. 133-138.

[30] M. Hosseini, K. Phalp, and R. Ali, "Towards crowdsourcing for requirements engineering," CEUR.org, vol. 1138, 2014.

[31] O. Barzilay, C. Treude, and A. Zagalsky, "Facilitating crowd sourced software engineering via stack overflow," Finding Source Code on the Web for Remix and Reuse. Springer, New York, 2013.

[32] S. Ramakrishnan and V. Srinivasaraghavan, "Delivering software projects using captive university crowd," in: Proc. International Workshop on Cooperative and Human Aspects of Software Engineering, IEEE, Hyderabad, India, 2014.

[33] T. Hossfeld, C. Keimel, and C.Timmerer, "Crowdsourcing quality-of-experience assessments," Computer, vol. 47, no. 9, pp. 98-102, 2014.

[34] A. Gritti, "Crowd outsourcing for software localization," https://www.welocalize.com, March 2015.

[35] T.D. LaToza, M. Chen, L. Jiang, M. Zhao, and A. von der Hoek, "Borrowing from the crowd: A study of recombination in software design competitions," in: Proc. International Conference on Software Engineering, IEEE, Florence, Italy, 2015, p. 551-562.

[36] M. Goldman, "Role-based interfaces for collaborative software development," in Proc. Symposium Adjunct on User Interface Software and Technology, ACM, Santa Barbara, USA, pp. 23–26.

[37] L. Tran-Thanh, S. Stein, A. Rogers, and N. Jennings, "Efficient crowdsourcing of unknown experts using bounded multi-armed bandits," Artificial Intelligence, vol. 214, pp. 89-111, 2014.

[38] W.-T. Tsai, W. Wu, and M. N. Huhns, "Cloud-based software crowdsourcing," IEEE Internet Comp., vol.18, no.3, pp.78–83, 2014.

[39] R. Kazman and H.-M. Chen, "The metropolis model a new logic for development of crowdsourced systems," Communications of the ACM, vol. 52, no. 7, pp. 76-84, 2009.

[40] W. S. Lasecki, J. Kim, N. Rafter, O. Sen, J. P. Bigham, and M. S. Bernstein, "Apparition: Crowdsourced User Interfaces that Come to Life as You Sketch Them," in: Proc. Conference on Human Factors in Computing Systems, ACM, Seoul, Korea, 2015, p. 1925-1934.

[41] T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, P. Tran-Gia, "Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing," IEEE Trans. Multimedia, vol. 16, 2014, p. 541–558.

[42] F. Guaiani and H. Muccini, "Crowd and laboratory testing can they co-exist?: an exploratory study," in: Proc. Int'l Workshop on CrowdSourcing in Sw Eng, IEEE, Piscataway, USA, 2015, p. 32-37.

[43] T. Xie, J. Bishop, R. N. Horspool, N. Tillmann, J. Halleux, "Crowdsourcing code and process via code hunt," in: Proc. International Workshop on CrowdSourcing in Software Engineering, IEEE, Piscataway, USA, 2015, p. 15-16.

[44] P. Minder and A.Bernstein, "Crowdlang: A programming language for the systematic exploration of human computation systems," Social Informatics. Springer, 2012.

[45] D. Naparat and P. Finnegan, "Crowdsourcing Software Requirements and Development: A Mechanism-based Exploration of 'Opensourcing," in: Proc. Americas Conference on Information Systems, Chicago, USA, 2013.

[46] D.Mehta, "An insight into software crowdsourcing: How crowd can transform the business model for technology service providers," Int'l Journal of Computer Applications, vol. 101, no. 12, 2014.

[47] T.W. Schiller, "Reducing the usability barrier to specification and verification," Dissertation, Universoty of Washington, USA, 2014.

[48] B. Gardlo, S. Egger, M. Seufert, and R. Schatz, "Crowdsourcing 2.0: Enhancing execution speed and reliability of web-based QoE testing," in: Proc. International Conference on Communications. IEEE, Sydney, Australia, 2014, p. 1070-1-75.

[49] J. Warner, "Next steps in e-government crowdsourcing," in: Proc. International Digital Government Research Conference, ACM, College Park, USA, 2011, p. 177-181.

[50] R. Kazman and H.M. Chen, "The metropolis model and its implications for the engineering of software ecosystems," in: Proc. FSE/SDP Workshop on Future of Software Engineering Research, ACM, Santa Fe, USA, 2010, p. 187-190.

[51] A. Dwarakanath et al, "CrowdBuild: a methodology for enterprise software development using crowdsourcing," in: Proc. International Workshop on CrowdSourcing in Software Engineering, IEEE, Florence, Italy, 2015, p. 8-14.

[52] L. Ponzanelli, A. Bacchelli, and M.Lanza, "Leveraging crowd knowledge for software comprehension and development," in: Proc. Conf on Sw Maintenance and Reeng., Genova, Italy, 2013, p. 57-66.

[53] M.A Storey, L. Singer, B. Cleary, F.F. Filho, and A. Zagalsky, "The (r) evolution of social media in software engineering," in: Proc. Future of Software Eng, IEEE, Hyderabad, India, 2014, p. 100-116.

[54] W. Ebner, JM. Leimeister, U. bretschneider, and H. Krcmar, "Leveraging the wisdom of crowds: Designing an IT-supported ideas competition for an ERP software company," in: Proc. Hawaii Int'l Conf. on System Sciences, IEEE, Waikoloa, USA, 2008, p. 417-417.

[55] R. Pham, L. Singer, and K. Schneider, "Building test suites in social coding sites by leveraging drive-by commits," in: Proc. International Conference on Software Engineering, IEEE, San Francisco, USA, 2013, p. 1209-1212.

[56] M. Zhao and A. van der Hoek, "A brief perspective on microtask crowdsourcing workflows for interface design," in: Proc. International Workshop on CrowdSourcing in Software Engineering, IEEE, Florence, Italy, 2015, p. 45-46.

[57] R. L. Saremi and Y. Yang, "Dynamic simulation of software workers and task completion," in: Proc. Int'l Workshop on CrowdSourcing in Software Engineering, IEEE, Florence, Italy, 2015, p. 17-23.

[58] M. Goldman, "Software development with real-time collaborative editing," Dissertation, Massachusetts Institute of Technology, 2012.

[59] T.D. LaToza, W.B. Towne, C.M. Adriano, and A. van der Hoek, "Microtask programming: Building software with a crowd," in: Proc. Symposium on User Interface Software and Technology, ACM, Honolulu, USA, 2014, pp. 43-54.

[60] B. Hartmann, D. MacDougall, J. Brandt, and S. Klemmer, "What would other programmers do: suggesting solutions to error messages," in: Proc. SIGCHI Conference on Human Factors in Computing Systems, ACM, Atlanta, USA, 2010, p. 1019-1028.

[61] T.D. LaToza, W.B. Towne, A. Van Der Hoek, and J.D. Herbsleb, "Crowd development", in: Proc. Int'l Workshop on Cooperative and Human Aspects of Sw Eng, IEEE, San Fco, USA, 2013, p. 85-88.

[62] M. Almaliki, N. Jiang, R. Ali, and F. Dalpiaz, "Gamified culture-aware feedback acquisition," in: Proc. Int'l Conference on Utility and Cloud Computing, IEEE-ACM, Shanghai, China, 2014, p. 624-625.

[63] K. T. Stolee and S. Elbaum, "Exploring the use of crowdsourcing to support empirical studies in software engineering," in: Proc. International Symposium on Empirical Software Engineering and Measurement, ACM-IEEE, Bolzano, Italy, 2010, Article 35.

[64] T.D. LaToza, W. B.Towne, and A. van Der Hoek, "Harnessing the crowd: Decontextualizing software work," in: Proc. International Workshop on Context in Software Development Workshop, 2014.

[65] Wei Li, Wei-Tek Tsai, Wenjun Wu, "Crowdsourcing for large-scale software development," Springer Berlin Heidelberg, 2015.

[66] E. Kalliamvakou, D. Damian, K. Blincoe, L. Singer, and D. German, "Open source-style collaborative development practices in commercial projects using GitHub," in: Proc. Int'l Conference on Software Engineering, IEEE, Florence, Italy, 2015, p. 574-585.

[67] K. Mao, Y. Yang, M. Li, and M. Harman, "Pricing crowdsourcing based software development tasks," in: Proc. Int'l Conf. on Software Engineering, IEEE, San Francisco, USA, 2013, pp. 1205–1208.

[68] K. Li, J. Xiao, Y. Wang, and Q. Wang, "Analysis of the key factors for software quality in crowdsourcing development: An empirical study on TopCoder.com," in: Proc. Computer Sw and Applications Conference Analysis, IEEE, Kyoto, Japan, 2013, pp. 812–817.