



# Crowd Agile Model for Effective Software Development

Shamaila Qayyum<sup>1(✉)</sup>, Salma Imtiaz<sup>1</sup>, and Huma Hayyat Khan<sup>2</sup>

<sup>1</sup> International Islamic University, Islamabad, Pakistan  
shamaillla@gmail.com, salma.imtiaz@iiu.edu.pk

<sup>2</sup> National University of Modern Languages, Islamabad, Pakistan  
humakhan0@gmail.com

**Abstract.** Crowd Sourced Software Development (CSSD) is becoming popular in software development industries due to reduced cost and efficiency. Many companies are moving towards crowdsourcing, and have already adopted Agile Software Development (ASD). However, CSSD differs from ASD in many ways due to its distributed nature. Although there is little research on the integration of these two approaches, whereas at the same time the combination of the two is advocated by some. It is deemed necessary to identify and resolve the issues emerged while integrating CSSD and ASD. This study hence intends to explore the issues emerged as a result of integrating agile and CSSD and propose a Crowd Agile model that will help in effective software development.

**Keywords:** Agile · ASD · Scrum · Crowd source software development · CSSD

## 1 Introduction

With the growth of software industry, traditional software development practices are getting old [1–3]. Internet and social media have provided ways for developers and employers to reach each other across the globe to get their task done [1, 3]. This approach of developing software which utilizes people around the globe for various kind of tasks is known as crowdsourcing [4, 5]. Recently Crowd Sourced Software Development (CSSD) has taken over the software industry [6]. The phenomenon involves outsourcing the tasks to the crowd consisting of unknown, heterogeneous people by an open call [4, 7]. The crowd, coordinates via any online platform [8] such as Amazon Mechanical Turk and Topcoder, to complete the tasks given by the employer [7]. This technique has said to improve the quality of tasks by increasing response time and reducing the cost [9].

CSSD enables the abilities of assorted individuals to be incorporated into a single venture [12]. It is commonly used for coding [13–16] and testing [17–23] in software development. In crowdsourcing, people performing tasks are heterogeneous gatherings, who do not know each other. As the software requirements are becoming unpredictable and researchers are emphasizing on more communication within team [24], CSSD is facing the challenges of team development [4, 9], developing volunteers' network [8],

task decomposition, trust among the crowd working on same project [14] and coordination.

On the other hand agile methodology is also a very popular software development methodology [25, 26] as it enables software teams to deliverable valuable services in flexible manner [27, 28]. It has been established that many companies adopting CSSD are also following ASD [29]. Companies, such as TPI, that use TopCoder as a platform, are shifting towards crowdsourcing and are already following agile methods [30]. The combination of crowd sourced software development and agile is one of an interesting and latest research area. As stated by Mishra [29], lately practitioners are following crowd source approach while working in agile environment, but research lacks sufficient articles on the effective integration of these two approaches. Stol [30] also highlighted the importance of effective combination of crowd and agile. The research is still very new and limited. Therefore, we aim to conduct an exploratory research to determine what are the challenges faced when CSSD is used within agile environment and what strategies can be followed to reduce these challenges, followed by an explanatory research in which we will validate our proposed model.

The rest of the document is divided into following sections; Sect. 2 gives the background on CSSD and agile. Section 3 provides literature review and recent state of art of agile and CSSD. Section 4 contains the objectives of this study and research questions this study intends to follow. Section 5 states detailed research methodology. Section 6 presents the contribution of this research and Sect. 7 provides a timeline in which this research will be completed.

## 2 Background and Motivation

In the recent years a new trend of software development has become popular in industry that lets the organizations involve isolated group of developers to develop software projects [1–3]. This concept is known as crowdsourcing. The term crowdsourcing was initially defined by Howe [10] in 2006. Social networking is one reason behind this globalization [1, 3]. A crowd can be any group of large number of anonymous people, which may comprise of experts as well as fresh graduates and inexperienced people. Crowdsourcing helps utilizing the skills of different people to be integrated into a single project [12].

At the same time, companies are widely adopting agile software development methodology [33]. Interestingly, some of the characteristics that are faced as challenges by CSSD are the benefits achieved as a result of following ASD. Agile not only keeps the team productive and motivated, but provides quality products [34]. ASD emphasizes on face to face communication [35, 36] and works in iteration with close collaboration of team, project manager and business people [27, 37]. Trust development among team members is another strength of ASD [38]. A daily meeting is held to discuss the progress of the project [34, 39]. ASD also helps in lowering cost of project as there is increased communication which decreases the chances rework and therefore cost overrun and project delay [24]. However, it is problematic executing CSSD in an agile environment [33].

Agile Software Development (ASD) has been adopted by the organizations doing Global Software Development (GSD) [27, 40–43]. GSD leads software developing companies, develop the software across remotely located teams [44]. Since many practices of agile are thought to solve different challenges of the organizations working across the globe [44–46]. CSSD is different from GSD, in GSD there are designated teams and CSSD comprise of a large group of unknown people who do not know each other neither are they designated employees of the organization [33]. Since agile practices can benefit GSD [33] we suggest they can be integrated and used during CSSD. How both can be effectively integrated needs to be explored [4]. This motivated us to conduct a research on exploring how agile and CSSD can be used together in a way that the maximum benefits of both the approaches can be achieved.

### 3 Related Work

#### 3.1 State of Knowledge on Crowd Sourced Agile Development

After reviewing recent literature we found a few studies on crowd sourced agile development. The most recent study among them on crowd source agile development was conducted by Klass Jan Stol [30] in 2019. They conducted interviews and found that the managers, who were involved in CSSD as well as agile, faced many problems because of these contradictory approaches. Researchers emphasized the need to investigate how these two approaches can be combined effectively. Likewise, another study by Klass Jan Stol is conducted in 2014 [4], where the researchers reported the significance of crowdsourcing. Mishra and colleagues in 2017 [29] conducted a study that reports the importance of integrating CSSD and ASD. The researchers highlighted the recent trends towards investigating agile development with crowd sourcing.

CSSD [1] and ASD [34] are widely adapted by the companies today since there are some key issues in following these both simultaneously as many characteristics of agile and crowdsourcing are contradicting each other. As [30] stated:

*“Given the widespread adoption of agile approaches to software development (in particular Scrum) that emphasize regular face-to-face communication, how can the crowdsourcing approach (which resembles a waterfall-style approach to software development with an emphasis on documented requirements) be effectively combined and coordinated?”*

Literature shows that ASD is also used by organizations doing GSD [40], [60]. Since GSD also has many characteristics that contradict with the principles of ASD. However, it has been established that agile practices are used for mitigating many challenges of GSD [44–46]. GSD differs from CSSD as in GSD there are designated teams but in CSSD a large group of unknown people is working on same task [33]. However, in CSSD the workers are globally dispersed as the designated teams in GSD are working from remote locations, so geographical difference is a commonality among both. If agile practices can be used for GSD, where teams are geographically apart from each other, it can also be used within CSSD for effective software development.

## 4 Research Questions

The study aims to explore the challenges of CSSD when used with ASD and the solutions that help reduce these challenges. This motivated us to design our research questions, as follows:

### RQ1:

How can Crowd Sourced Software Development (CSSD) be effectively integrated with Agile Software Development (ASD) to achieve maximum benefits of both?

*RQ 1.1:* What are the challenges when ASD is used with CSSD?

*RQ 1.2:* What strategies could be used to overcome the challenges of ASD and CSSD integration?

*RQ 1.3:* How can the crowd agile model help CSSD teams be more successful in their use of ASD?

## 5 Methodology

To answer each research question of the study, following research methodology will be adapted. Table 1 gives a detailed research summary along with the methodology.

**Table 1.** Research summary

Research question	Objective	Methodology	Outcome
RQ 1.1	To identify the challenges of integrating CSSD and ASD from literature as well as industry	- Literature review - Survey	List of issues faced by companies following crowd source approach with agile software development
RQ 1.2	To identify the practices that can address the identified challenges of CSSD and ASD	- Literature Review - Survey	- List of solution strategies adapted by companies to resolve issues while using crowd sourcing approach within agile environment
	To propose a Crowd Agile model		- a model based on the issues and solutions identified in previous phases
RQ 1.3	To evaluate how crowd agile model helps in effective software development	- Case study - Plan B - Experiment	An evaluated model

*Literature review (LR)* will be conducted in order to find the reported issues while working together with crowd sourcing and agile. LR will also be carried out to find the solutions and strategies for the issues emerged that are given in the literature.

A *survey* will be conducted to find out what issues practitioners are facing when they use crowdsourcing while working in an agile environment. This survey will also

figure out what strategies these practitioners adopt to overcome the issues faced by combining crowdsourcing with agile. The survey will be *exploratory* as we will be identifying issues and the strategies that are faced by industry. Guidelines for conducting survey by Kasunic [47] will be followed. *Type of survey*: This survey will be self-administered. Self-administered surveys are those which are sent online through web based surveys or emails. The survey will be a mix of both open and closed questions. *Target audience*: The targeted audience for our survey is the software development practitioners and managers working with crowdsourcing and agile. The target audience will be approached by different platforms which allow crowdsourcing. It will be first confirmed through email/any other contact medium that they are also following agile.

*Case study*: The evaluation of the model proposed in this study, will be done with a case study. For carrying out case study effectively, guidelines for conducting and reporting case study research in software engineering by Runeson [48] will be followed. Case study will be *explanatory*. This study intends to find how CSSD and agile can be integrated and what benefits can be achieved from this integration. *Case definition*: The case of any software development company will be considered who carry out software engineering activities through crowdsourcing while working in agile environment. *Unit of analysis*: Our unit of analysis will be the software project and the practitioners that will develop software using our model. *Data Collection*: Data collection from the case study will be done through second level technique. The researchers will directly collect raw data by monitoring the use of the model. This will be done using *interviews*. *Data analysis*: Qualitative data analysis will be done. Narrative analysis and discourse analysis will be used for this purpose. As the data will be received in the form of different responses from the respondents” so it will be better analyzed in the form of respondents’ stories, talks and texts.

We understand the issues in identifying an organization for case study, as a plan B, we will conduct an experiment for the evaluation of our proposed model.

## 6 Research Framework

*Theoretical Perspective of Research*: The theoretical perspective of this study is based on organizational theory as it focuses on the improvement in organizational performance. A positivist approach is being followed as the research aims to study human behavior and results are produced on empirical basis.

*Methodological Perspective of Research*: This is *applied research* as the sole purpose of carrying out this research is to address the problems of an organization which is using ASD along with CSSD. The type of applied research this study follows is *evaluation research*, as we intend to study the impact of our suggested model.

## 7 Expected Outcomes

The outcome of this study will be in the form of a model. In this model guidelines will be provided for effective use of CSSD with ASD. These guidelines will be in the form of challenge-strategy pair. This pair will specify that for any particular challenge (while integrating CSSD with ASD), what strategy can be followed so that the benefits of both can be achieved. The contribution of this research will be; the identification of issues while using CSSD with agile, the identification of solutions for the issues identified while using CSSD with agile, a crowd agile model for effective usage of CSSD and agile and validation of the model.

## References

1. Begel, A., Bosch, J., Storey, M.I.: Social networking meets software development: perspectives from github, msdn, stack exchange, and topcoder. *IEEE Softw.* **30**(1), 52–66 (2013)
2. Tamburri, P., Vliet, H.: Organizational social structures for software engineering. *ACM Comput. Surv.* **46**(1), 1–35 (2013)
3. Storey, M.A., Singer, L., Filho, F.F., Zagalsky, A., German, D.M.: How social and communication channels shape and challenge a participatory culture in software development. *IEEE Trans. Softw. Eng.* **41**(7), 185–204 (2015)
4. Stol, K.J., Fitzgerald, B.: Two's company, three's a crowd: a case study of crowdsourcing software development. In: *Proceedings of the 36th International Conference on Software Engineering*, pp. 187–198 (2014)
5. Ågerfalk, P.J., Fitzgerald, B., Stol, K.J.: *Software Sourcing in the Age of Open*. SCS. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-17266-8>
6. Mao, K., Capra, L., Harman, M., Jia, Y.: A survey of the use of crowdsourcing in software engineering. *J. Syst. Softw.* **126**, 57–84 (2017)
7. Mooty, M., Faulring, A., Stylos, J., Myers, B.A.: Calcite: completing code completion for constructors using crowds. In: *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 15–22 (2010)
8. Schenk, E., Guittard, C.: Towards a characterization of crowdsourcing practices. *J. Innov. Econ. Manag.* (1), 93–107 (2011)
9. Hosseini, M., Phalp, K., Taylor, J., Ali, R.: *Towards crowdsourcing for requirements engineering* (2014)
10. Howe, J.: *The rise of crowdsourcing [Электронный ресурс]* (2006)
11. Brabham, D.: *Crowdsourcing*. MIT Press, Cambridge (2013)
12. Vander Schee, B.A.: Crowdsourcing: why the power of the crowd is driving the future of business. *J. Consum. Mark.* **26**(4), 305–306 (2009)
13. Latoza, T.D., Zhao, M., Van Der Hoek, A., Chen, M., Jiang, L., Van Der Hoek, A.: *Borrowing from the Crowd: A Study of Recombination in Software Design Competitions Microtask Programming View Project Merge Nature View Project Borrowing from the Crowd: A Study of Recombination in Software Design Competitions* (2015)
14. Oi, D., Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing systems on the worldwide web. *Commun. ACM* **54**(4), 86–96 (2011)
15. Morisaki, S., Usui, Y.: An Approach for crowdsourcing software development. In: *Proceedings Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement*, pp. 32–33 (2011)

16. Xu, X., Wang, Y.: Crowdsourcing software development process study on ultra-large-scale system. In: *Advanced Materials Research*, pp. 4441–4446 (2014)
17. Schneider, C., Cheung, T.: The power of the crowd: performing usability testing using an on-demand workforce. In: *Information Systems Development*. Springer, New York (2013). [https://doi.org/10.1007/978-1-4614-4951-5\\_44](https://doi.org/10.1007/978-1-4614-4951-5_44)
18. Liu, D., Bias, R.G., Lease, M., Kuipers, R.: Crowdsourcing for usability testing. *Proc. Am. Soc. Inf. Sci Technol.* **49**(1), 1–10 (2012)
19. Nebeling, M., Speicher, M., Grossniklaus, M., Norrie, M.C.: Crowdsourced web site evaluation with crowdstudy. In: Brambilla, M., Tokuda, T., Tolkdorf, R. (eds.) *ICWE 2012*. LNCS, vol. 7387, pp. 494–497. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-31753-8\\_52](https://doi.org/10.1007/978-3-642-31753-8_52)
20. Meier, F., Bazo, A., Burghardt, M., Wolff, C.: Evaluating a web-based tool for crowdsourced navigation stress tests. In: Marcus, A. (ed.) *DUXU 2013*. LNCS, vol. 8015, pp. 248–256. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39253-5\\_27](https://doi.org/10.1007/978-3-642-39253-5_27)
21. Speicher, M., Nebeling, M., Norrie, M.C.: CrowdStudy: general toolkit for crowdsourced evaluation of web interfaces In: *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, pp. 255–264 (2013)
22. Teinum, A.: User testing tool towards a tool for crowdsource-enabled accessibility evaluation of websites (2013)
23. Gomide, V.H.M., et al.: Affective crowdsourcing applied to usability testing. *Int. J. Comput. Sci. Inf. Technol.* **5**(1), 575–579 (2014)
24. Bowes, J.: Kanban vs scrum vs XP—an agile comparison. *Kanban vs Scrum vs xp* (2015)
25. Williams, L., Cockburn, A.: Underlying values (2003)
26. Boehm, B.: Get ready for agile methods, with care. *Computer* **35**(1), 64–69 (2002)
27. Hossain, E., Ali Babar Lero, M.A.: Using scrum in global software development: a systematic literature review Hye-young Paik. In: *2009 Fourth IEEE International Conference on Global Software Engineering*, pp. 175–184 (2009)
28. Sutherland, J., et al.: Distributed scrum: agile project management with outsourced development teams. In: *2007 40th Annual Hawaii International Conference on System Sciences (HICSS 2007)* (2007)
29. Mishra, A., Garbajosa, J., Wang, X., Bosch, J., Abrahamsson, P.: Future directions in agile research: alignments and divergence between research and practice. *J. Softw. Evol. Proc* **00**, 1–5 (2017)
30. Stol, K.J., Caglayan, B., Fitzgerald, B.: Competition-based crowdsourcing software development: a multi-method study from a customer perspective. *IEEE Trans. Softw. Eng.* **45**(3), 237–260 (2019)
31. Latoza, T.D., Ben Towne, W., Van Der Hoek, A., Herbsleb, J.D.: Crowd development. In: *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pp. 85–88 (2013)
32. Greengard, S.: Following the crowd. *Commun. ACM* **54**(2), 20–22 (2011)
33. Li, W., Tsai, W.T., Wu, W.: Crowdsourcing for large-scale software development. In: Li, W., Huhns, M.N., Tsai, W.T., Wu, W. (eds.) *Crowdsourcing*. PI, pp. 3–23. Springer, Heidelberg (2015). [https://doi.org/10.1007/978-3-662-47011-4\\_1](https://doi.org/10.1007/978-3-662-47011-4_1)
34. Profile, S., Alyahya, S., Alsahli, A., Khan, H.: Agile development overcomes GSD challenges: a systematic literature review. *Int. J. Comput. Sci. Softw. Eng.* **6**(1), 7 (2017)
35. Newkirk, J.: Introduction to agile processes and extreme programming. In: *Proceedings of the 24th International Conference on Software Engineering*. ICSE 2002, pp. 695–696 (2002)

36. Damian, D., Lassenius, C., Paasivaara, M., Schröter, A., Borici, A.: Teaching a globally distributed project course using Scrum practices NaPiRE: Naming the Pain in Requirements Engineering View project Need for Speed View project Teaching a Globally Distributed Project Course Using Scrum Practices (2012)
37. Srivastava, A., Bhardwaj, S.: SCRUM model for agile methodology. In: 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 864–869 (2017)
38. Kniberg, H., Skarin, M.: Kanban and Scrum-Making the Most of Both. Lulu. com, Morrisville (2010)
39. Rubin, K.: Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley, Boston (2012)
40. Holmström, H., Fitzgerald, B., Ågerfalk, P.J., Conchúir, E.Ó.: Agile practices reduce distance in gloral software development. *Inf. Syst. Manag.* **23**(3), 7–18 (2006)
41. Beecham, S., Noll, J., Richardson, I.: Using agile practices to solve global software development problems - a case study. In: Proceedings International Computer Software and Applications Conference, 18–21 August 2014, pp. 5–10 (2014)
42. Hossain, E., Babar, M.A., Verner, J.: How can agile practices minimize global software development co-ordination risks? In: O’Connor, R.V., Baddoo, N., Cuadrado Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) EuroSPI 2009. CCIS, vol. 42, pp. 81–92. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-04133-4\\_7](https://doi.org/10.1007/978-3-642-04133-4_7)
43. Shrivastava, S.V., Date, H.: Distributed agile software development: a review, June 2010
44. Herbsleb, J.D.: Global software engineering: the future of socio-technical coordination, in FoSE. *Future Softw. Eng.* **2007**, 188–198 (2007)
45. Bannerman, P.L., Hossain, E., Jeffery, R.: Scrum practice mitigation of global software development coordination challenges: a distinctive advantage?. In: Proceedings of the Annual Hawaii International Conference on System Sciences, pp. 5309–5318 (2012)
46. Hossain, E., Bannerman, P.L., Jeffery, D.R.: Scrum practices in global software development: a research framework. In: Caivano, D., Oivo, M., Baldassarre, M.T., Visaggio, G. (eds.) PROFES 2011. LNCS, vol. 6759, pp. 88–102. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21843-9\\_9](https://doi.org/10.1007/978-3-642-21843-9_9)
47. Kasunic, M.: Designing an Effective Survey DISTRIBUTION STATEMENT a Approved for Public Release Distribution Unlimited (2005)
48. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **14**(2), 131–164 (2009)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

