The Good, the Bad and the Ugly: An Onboard Journey in Software Crowdsourcing Competitive Model

Leticia Machado,
Alexandre Zanatta
Computer Science School, PUCRS
Porto Alegre, Brazil
[leticia.machado.001,alexandre.zanatta]@acad.pucrs.br

Sabrina Marczak,
Rafael Prikladnicki
Computer Science School, PUCRS
Porto Alegre, Brazil
[sabrina.marczak, rafaelp]@pucrs.br

Abstract—**This paper reports on a study that aimed to characterize how crowd workers experienced for the first time the use of TopCoder, a crowdsourcing platform for software development that implements a competitive model. We explored how they perceived collaboration in this setting, what challenges they faced to perform a single task, and reflect upon their suggestions to overcome the challenges their experienced. More specifically, we asked graduate students to select a development challenge task, work on it, and submit their contribution to the platform. Early analysis of the results: (1) reveal the potential benefits of software crowdsourcing from the crowd perspective, (2) discuss collaboration in a competitive model, and (3) highlight that the onboarding process for newcomers is seen as challenging. We discuss our findings in light of current literature.**

Keywords-component: crowdsourcing, software development competitive model, TopCoder, newcomers

I. Introduction

Software Crowdsourcing (SW CS) refers to the act of externally transfer any task of the software development process (from a requester) to a potential and undefined large group of online workers—the crowd, in an open call format [1] through a digital platform. This takes place based on distinct crowdsourcing models, namely: peer production, competition, and microtasking [2].

Software development models based on the crowd through competition and microtasking have drawn more attention of companies, autonomous developers, and startups since they display a flexible format to tap on demand in the IT area by decomposing tasks into short, self-contained pieces of work that can be independently and quickly performed [3]. This movement suggests a trend similarly to what happened with the Distributed Software Development fifteen years ago [4]: a new business/development model has come to stay.

Competition is the model in which this study focused on. It consists in having crowd workers independently creating a solution, competing against each one by anchoring a monetary reward for its task completion [5].

An important limitation of current SW CS competitive model is the collaborative relations among involved parties (crowd, requerters and platform). Kotlarsky and Oshri [14] argue that collaboration is a complex and multidimensional process characterized by communication, coordination, interaction, relationship, trust, and structure aspects. In SW CS, collaboration can feature peculiar characteristics and structures due to the design of work, i.e., work split into tiny bits, which can be disseminated, claimed, and performed by people worldwide in an individual basis.

There are several studies discussing software crowdsourcing, but only a few provide empirical evidence. For instance, several concerns regarding the adoption of SW CS from the requester's perspective using TopCoder are reported by [3]. A requester representative stated that the related concerns demand from the requester effort to prepare specifications and answer crowdsourcing community queries, review submissions, and resolve coordination and quality issues during SW CS projects. More broadly, LaToza and Hoek [2] and Stol and Fitzgerald [3] reported that key challenges still remain for SW CS to reach its true potential. These challenges are: crowd labor, task decomposition, quality assurance, coordination, collaboration and communication, motivation and remuneration, and knowledge and intellectual property.

Bearing in mind that SW CS is an emergent area, there is still little information about what challenges crowd developers face during the task execution and to what extent collaboration activities among involved parties are present during SW CS development that operates on a structure of competitions.

Understanding collaboration activities and challenges faced by crowd developers while trying to submit solutions to SW CS projects using for the first time TopCoder motivated this paper. We present and discuss our findings on a student-based study in this paper.

II. Topcoder

Topcoder [6] is one of the main platforms for competitive SW CS projects worldwide [7]. It covers all phases of the life cycle, from elicitation to deployment, in which for each of these phases an open call to the crowd is made [5]. The open call is composed of six steps [8], namely: posting, registration, submission, review, appealing, and winner. On Topcoder, there is a mediator between the platform and the crowd to follow up on the tasks development, named copilot. The copilot may be a member of the crowd playing the representative role of the client who demanded the task. The copilot's main responsibilities are related to managing questions submitted by crowd members in task-based forums and answering them, updating information, and so on. It is possible for participants of the task to open new threads within the forum and reply to messages from anyone (crowd and copilot) who is active in the forum.

The platform offers services expressed in the form of tasks organized into categories of challenges as follows: *Design Challenge, Development Challenge, Data Science Challenge*, and *Competitive Programming* [6]. A task in the Development Challenge category is categorized in one of the following sub-categories: Architecture, Assembly, Bug hunt, Code, Concept, Content creation, Copilot, Component design, Component development, First to Finish (F2F), Marathon, RIA, Spec, Test scenarios, Test Suites, or UI prototype [6]. Despite the category, each task has a scope and is composed of technical requirements that define the expected behavior of that software task and the necessary interfaces to integrate with other parts of the system.

For development challenge-based tasks, TopCoder has recently started to make available UML tools, such as sequence, class, use case, and activity diagrams, where members could check to support the development solution, in addition to the task description itself.

Relevant information, such as requirements of the SW CS task, schedule, budget, among others, are negotiated by the requester itself, which manages the platform directly with the requester who demanded the task on Topcoder.

III. Research Setting

Our study took place as part of a graduate course project on Collaborative Software Development (CSD) at PUCRS, Brazil. The course discusses the history of collaborative systems and CSCW principles as well as different models of collaborative software development including global software engineering and software crowdsourcing. The course is offered once a year and has, in average, about 15 students enrolled per session.

The 16 weeks-long course consists of lectures, group discussions and presentations of selected papers, and a final project. At this course session, the final project aimed to promote the discussion of collaborative software development considering the software crowdsourcing competitive model. We selected TopCoder as the platform for the project to take place in. In Week 4 the students were introduced to the final project and given 2 weeks (Week 6) to select a task of their choice from the Development Challenge (DC) category to work on.

The students were told they had to work independently and that they has 6 weeks to conclude the task (Week 12) and submit it in the platform, and 2 additional weeks (Week 14) to observe and respond to any feedback, if given, from the platform. A class discussion was hold in Week 15, and feedback and grades reported back to the students in Week 16. An assignment was associated to each of the above-mentioned activities as follows:

a) Report 1 - Task Description (Week 6): the student reported on the selected task and provided us with a brief description of the task goal, sub-category within the Development Challenge category, and estimated period of completion. A brief explanation on the reasons of selecting this task was also provided;
b) Report 2 – Open experience report (Week 12): the student had to respond to the 9 open questions asked about their experience on using TopCoder for the first time. The questions that enabled students to debrief and explain their SW CS experience in terms of collaboration activities and barriers about the onboarding process while trying to place their code solution on Topcoder;
c) Report 3 – Open-ended online questionnaire (Week 14): this questionnaire provided us with profiling information on the students' background and Likert-scale based questions asking about their opinion on the course project.

This paper reports on the qualitative data collected in Report 2. Data was analyzed using Content Analysis, as described by [9]. It is important to note that students were given the change to develop the course project and not sign the research consent form. All students voluntarily agreed to join the study.

IV. Results

In this section, we present our initial analysis of the results regarding the perceptions about the participants' experience, collaboration activities, and challenges in SW CS projects conducted in the case study.

The group of 21 students mostly participated in a SW CS initiative for the first time, that is, they submitted to a SW CS task on Topcoder for the first time, whereas some of them had already submitted SW CS tasks' solution in other platforms such as Upwork (http://www.upwork.com), and People per Hour (http://www.peopleperhour.com), besides Topcoder. The graduate students were experienced developers with an average of over 6 years of experience, ranging in age between 26 and 30 years old.

The students took part in "F2F" and "Code" tasks in TopCoder, which means that only these two types have been chosen from DC category.

A. Experience in SW CS Project

The question related to the participants' experience was: as follows "**What aspects did you find interesting regarding your experience in software crowdsourcing with Topcoder platform?"**

Grasping the participants' experience in this case study may help us better understand some of the characteristics and difficulties of SW CS model on Topcoder platform. All identified aspects were also categorized as positive or negative. For this categorization (positive or negative) we used the syntactic analysis, identifying adjectives or adverbs that could indicate the polarity of the content.

The main characteristics of SW CS model were reported as positive aspects of the first experience in SW CS with the use of Topcoder platform. Examples of key aspects are the freedom to choose the task according to one's profile (expertise demands), as reported by P11:*"I found it interesting that the platform allows one to select any activity that one consider suitable to develop."*

The division of tasks into small bits and the possibility to offer several solutions to the same problem were highlighted by P8: *"I saw that it is a model that enables the client to obtain the best solutions."*

It is also associated to "extra" earnings (money-wise and incentives), as participant P4 reports: *"I see it as an opportunity for those [developers] who wish to earn some extra money."*

Activities related mainly to task management, such as devoting a great effort to perform the task and staying motivated to perform the task given that one knows upfront that only a person or two will win the challenge were aspects reported as negative.

Despite some difficulties in executing the task in an extremely competitive environment, the experience faced by participants in the SW CS model on the Topcoder platform was considered positive. The account made by P6 reinforces such perception*: "The experience was incredible, because I had no idea of how this platform really was used and how it offered opportunities for independent developers. I might continue using the platform to look for opportunities."*

It is interesting to note that, despite being their first participation in SW CS activities on TopCoder, many participants highlighted features of the crowdsourcing model such as diversity per task, extra money and alternative solutions for the same problem.

B. Collaboration Perception in SW CS Projects

The question related to participants' collaboration perception was as follows: **"How did you perceive collaboration in SW CS on Topcoder among crowd members and between the platform (requester) and its members?"**

The collaboration perception in SW CS on the platform and the task execution sought to investigate how and when the involved parties (crowd, platform and requester) collaborate in the work setting. As a consequence, this feedback will help us understand to what extent SW CS competitive model allows and requires collaboration during task execution among members contributing to the same software product development.

Findings suggest that the collaboration perception in SW CS competitive model was fuzzy, which was gathered through three answers given by them: those who perceived collaboration, those who partially perceived it, and the ones who did not perceive collaboration while participating in the task execution on Topcoder.

Referring to the answers of those who perceived collaboration clearly or partially, that is strongly connected to message exchange and seeking information about the task on communication forums.

Regarding those participants who reported not noticing collaboration in SW CS, it was observed that this fact is intrinsically bound to two main reasons: (i) not working together with other members to write code, and (ii) not using the forum to communicate during task development.

Considering that the goals in SW CS development are to obtain redundant solutions for the same problem (task) submitted by the largest number of participants, such solution diversity in SW CS contrasts to the software development process carried out internally by companies, as well as Open Source Software [2]. In these models, developers discuss feasible and diverse solutions to a given software problem and make decisions on developing a solution that shall be then performed by all.

It is noteworthy that, through the patterns found in the analyzed data, the forum is seen as a synonym of collaboration in the task development process according to crowd participants. Thus, the forum – considered as an asynchronous tool of communication among task participants (crowd), and between the task mediator (*co pilot*) and the crowd – represents the most used means to share task specification documents and exchange information.

Furthermore, narrowing down to Topcoder, forums are used as a task communication and coordination resource and they are just visible to those participants who have registered to the task. From this moment onward, participants can post messages or simply read and gather information from the posts of other task participants.

For most tasks performed in the study, participants reported the "presence" of a copilot in the forums, which may justify the responses/feedback from participants about the perception of collaboration in a more explicit and direct way between the copilot and crowd to refine the scope and support doubts about task documentation.

For most participants, the perception of collaboration among crowd members during the performance of the task was weak, indirect, and not clear/visible, as can be seen by some participants' responses: *P20: "I only noticed* *collaboration between client and developer through the forum. And I think this is the downside of the model adopted by Topcoder. As the work is conducted as a competition, collaboration among members is minimized."*

We preliminarily coded the messages (conversations) from the forums into communication topics (Table 1), and coordination topics (Table 2), because they were the mentioned messages related to collaboration aspects, from the data reported by the participants. Moreover, we have also separated them by messages exchanged between copilot and crowd, and between crowd and crowd.

TABLE 1: MESSAGE CONTENT BETWEEN CO PILOT AND CROWD ORGANIZED BY COMMUNICATION TOPICS

Communication Topics	
Id.	***Messages***
1	Share new task information (links, reference materials for task development
2	Provide task documentation such as: downloadable files, screenshots / screens, images, diagrams, etc
3	Solve understanding problems on documentation
4	Anticipate crowd issues and doubts
5	Clarify questions about task requirement details
6	Communicate useful information to perform the task
7	Provide clarifications and restrictions of task specification

TABLE 2: MESSAGE CONTENT BETWEEN CO PILOT AND CROWD ORGANIZED BY COORDINATION TOPICS

Coordination Topics	
Id.	***Messages***
1	Provide solution for possible document failures linked with the task
2	Make decisions on task correction
3	Manage complaints about task complexity versus remuneration
4	Reminder of task time / deadline

1) Co pilot interventions: messages posted by co pilot on the forum during the execution of the tasks.
2) Crowd Interventions: messages posted by crowd participants during the development of the task.

- notify errors and inconsistency in the task documentation provided by the requester
- exchange tips on locating task-related documents (files)
- share useful information to perform the activity
- request further detailing on the task

For example, participant P6 reported: *"Collaboration was undoubtedly essential and it worked objectively. Specifically in the activity I chose, I did not find the files to download and I searched for information in the task Forum. Not only did I find the files but I realized that other colleagues had already identified problems in the files and had exchanged information that was useful for the execution of the task.*"

In addition, another participant's feedback states: *"P12: Collaboration among members of Topcoder platform is non-existent; identifying other members on the platform is very difficult. Between the client / platform and the user, there is certain collaboration through a forum that the platform offers, but this collaboration is more used for task documentation and clarifying doubts about some items of the task that were not clear."*

A smaller number of interventions and forum content among members of the task may have been influenced by the context of Topcoder's online contest (competition), financial reward, and time to be the first to send the best solution, as mentioned by some participants in this study.

The e-mail tool was perceived as collaboration between platform and crowd, cited by only one of the participants. E-mail was mentioned mainly to the post submission review process of the solution developed for the task. As mentioned by P14: Among platform members: *"Collaboration on the platform, as far as I have noticed, works through messaging forums, where developers ask direct questions to the requester and the answers can be viewed by everyone who is attending that particular task and can also make use of that information".*

Between the platform (or client) and platform members: *"The information that the platform makes available to the developer refers to reviews the task is going through and the end time. All other information I received from the platform was through e-mails with information about the process, but most of the time emails were repeated".*

C. Challenges in the SW CS Project

The question related to participants' challenges was as followed: "**What other types of difficulties did you find while participating in the platform? Difficulties may be of a personal and / or technical nature, both in the use of the platform and in the accomplishment of the task".**

During the process, we analyzed the answers, identified barriers in the platform user's experience, and tasks execution emerged from this analysis. The identified challenges are listed below ordered from most frequent to least frequent:

Platform usability: ISO 9241-11 defines usability as *"the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*". Regarding the perception of effectiveness, some participants commented that it is very difficult to complete the task, which requires much effort. In addition to that, with regards to efficiency issue, our participants commented that it is very difficult to find some information about the tasks. A participant stated: *"P10: When using the platform, I had trouble finding the files that the client would like to see rectified."*

Another participant mentioned: *"P13: Initially, I found a little difficulty in finding the material available to perform the task."*

Task management: Basically the main difficulties encountered by the participants relate to finding a task according to their profile, understanding it and allocating personal time to do it. After choosing the task according to the participant's profile, as a participant commented *"My main difficulty was finding a task with the technologies I master."* (P8)

It calls attention the lack of documentation or incomplete documentation related to the execution of the task, as another participant commented: *"The documentation provided was not enough to fully understand what was supposed to be developed."* (P9)

Documentation associated with the accomplishment of the task is fundamental for its effective execution. Difficulties with documentation may be linked, in a way, to many problems of assembly environment setup for the execution of the task. P1 commented: *"I initially had difficulty to setup the environment."*

Another participant reinforced this aspect by saying: *"Assembling the right environment to carry out the task was always a challenge."* (P7)

Besides all difficulties in finding a task with the participant's profile and setting up the environment for the execution of the task, participants also need personal time available for the execution of the task within the time frame stipulated by the requester. P8 reported: *Another difficulty was finding a task that I could solve within the time window set up to hand the report out."*

Feedback: One difficulty reported by the participants was the delay in receiving feedback to a submitted task, as reported by P13*: "I missed getting partial feedback during the time the challenge had been active."*

Or simply not receiving feedback on the task sent, as commented by P1*:"The second point was the review process as a whole, in which I was hoping to receive feedback from the written code and then be able to change the code and submit it again, but that did not happen.*"

In addition to that, another participant commented: *"I also found that there was lack of better feedback after submitting the solution."* (P5)

Feedback is a feeding process that occurs through replying by means of criticism, evaluations, suggestions or comments about a performed activity, whose purpose is to aid in the process of possible performance adjustments or in the performance of the individual who carried out the activity. In a model such as SW CS, which depends on collaborative tasks among participants, feedback is paramount to the maintenance of these participants in this model [10].

One participant commented: *"During each review, the user is notified by email at its end, but without a breakdown, progress report or performance result in the activity, this feedback helps in the professional improvement of the user."* (P18)

It is worth remembering that feedback is not an opinion that expresses an emotion, but rather something that feeds, validating or not, achievements based on clear, objective and verifiable parameters.

In view of the above, it is suggested that all sent tasks from code and F2F categories of Topcoder platform shall be evaluated by the platform and that participants shall be informed about this evaluation. It is important for this practice of feedback to be systematic and constructive. P18 commented: *"feedback by email, explaining the sort of review that is ongoing and the outcome of this review to the user is essential to knowledge."*

D. Suggestions to minimize challenges in SW CS projects

The question related to participants' suggestions to minimize challenges were: **"Which suggestions do you propose to minimize these difficulties?**

The suggestions mentioned by the participants are based on the difficulty to seek information about the task during execution as well as bringing developer participants onboarding on the Topcoder platform, and lack of interactive feedback.

1) *Suggestions for Platform usability*
   - Make the layout of the review system more integrated with the dashboard
   - Standardize needs related to location and download of task files
   - Improve patterns of interaction with other users
   - Greater transparency and interaction in the workflow of the task (e.g. status of each step)
2) *Suggestions for Task management*
   - Indicate development environment for a task
   - Integrate tasks with other development tools and collaborative repository (Github, Stackoverflow, etc.)
   - Update the documentation of task requirements changed in the forum
   - Better divide and describe tasks due to high granularity
   - Recommend new tasks based on participants' profile
3) *Suggestions for Feedback*
   - Improve code review process
   - Provide a synchronous communication channel

It is worth remembering that all suggestions presented by the participants refer to activities performed just in the "Code" and "F2F" sub-categories of Development Challenge category in Topcoder, and were not validated.

V. Discussion

A. Individual behavior vs. collaborative behavior

Open calls for competing solutions in SW CS projects apply a model of independent and individual work that can inhibit the opportunity for developers of the crowd to collaborate in exchanging information, ideas and so on. It was observed in preliminary results that some SW CS activities are performed individually by crowd members, i.e., individual behavior does not require other developers' participation (e.g., coding, reading documentation, onboarding task process), whereas collaborative behavior requires more than one person to be involved to interact for instance, making decision, informal conversations, post submissions in forums, chat in real time communication, and e-mail messages.

Thus, during the execution of SW CS tasks, the collaboration takes on new characteristics in this competitive model, influenced by the transitory, remote, disperse, unknown/anonymous aspects of the crowd, making it easier to recognize individual behavior rather than collaborative behavior among platform members who share an award for the best submitted solution as referenced by Machado et al., [11].

As mentioned by Herbsleb and Grinter [4], global distance affects collaboration in the process of software development and given that SW CS development operates on a structure of competitions [5], these can impose a restriction to crowd members collaborate with one another. In this way, we were surprise that participants reported about collaboration during their experience in SW CS tasks on Topcoder. Thus, new aspects to coordinate distributed individuals online and manage the flow of communication emerge in a competitive SW CS context: Shall we expect collaboration? Shall we reconsider what we know? Tasks are being defined to a single developer to work on her own, but is this possible?

B. *On the usage of asynchronous communication to interact and collaborate*

Informal communication among crowd members during tasks' execution is restricted to asynchronous tools, according to what was observed in this study. Asynchronous communication via forum in each task performed by the participants of the study reinforces that interaction among crowd members and copilot may impact collaboration for the development of the task. We may think that the platform intentionally supports communication among members of the crowd in a restricted way, by considering the competitive model it acts. No direct communication during the development of SW CS tasks may lead to misunderstandings and thus may have a big impact on the quality of the final solution [3]. Asynchronous interactions do not provide rapid feedback among members. For instance, in forums, feedback communication is delayed and interruptions or long pauses in communication often occur [10]. However, forum has a considerable influence on the interaction for communication during a SW CS task execution on Topcoder, as we have found in the results. In connection to the usage of asynchronous communication to collaborate and interact in this platform, it becomes possible to think if forum resource in competitive SW CS represents a particular collaboration way among the involved parties (crowd, requester and platform) to development software tasks in this context.

Moreover, as mentioned before [3], communication and coordination concerns can impose an overload during SW CS projects in terms of the effort to prepare task specification documentation, significant amount of communication to answer crowd questions, evaluation and feedback to submitted solutions by crowd participants.

C. *Onboarding process and user experience.*

Platform usability, task management and feedback were pointed out as the main challenges in SW CS of Topcoder platform also mentioned in Zanatta et al. [12].

The experience of participating in a new approach to software development evidenced an onboarding challenges in SW CS, especially with regard to the design of work - project split into tasks, seeking information, isolated and individual actions among involved parties (requester, crowd and platform), environment setting, and review process that disappointed the crowd who submitted their solution to the platform. These challenges pose some questions: Are the reports related to the lack of information available? Is this a platform design limitation?

In relation to the review process, developers must be privately notified by the platform in order to provide greater transparency and interaction on the weaknesses and strengths of the submitted solution. It is important that this feedback practice for the review and appealing steps adopted by Topcoder [7] is systematic and constructive [13].

A participant commented: *"Feedback by e-mail, explaining the type of review that is ongoing and its result to the user is fundamental to knowledge.*" (P18)

According to LaToza and Hoek [2] "a key tenet of crowdsourcing is that each participant must be precisely informed of the task to be performed", and based on our results, it is essential to provide complete, consist and clear documentation about the tasks, because this can help motivate practitioners to improve the overall platform usability.

I. VII. Conclusions and Future Work

As shown in the previous sections, our goal in this paper is to understand and characterize crowd workers' experience with software development in a significant context such as SW CS competitive model. Besides that, we investigated how developers collaborated with each other in a work setting, and what the main challenges faced by them were in this context.

As contribution, some preliminary suggestions to the challenges in SW CS under the crowd perspective are provided through data analysis. Besides that, these suggestions should provide subsidies to improve the platform requirements to be more assertive in setting collaboration aspects and onboarding processes in SW CS projects. We also find that the review process of Topcoder imposed a significant difficulty, which is little mentioned in the literature of the field.

The positive experience and the unexpected collaboration between crowd competitors can be mentioned as good aspects in the SW CS onboard process. On the other hand, the need to improve the general usability of the platform, adding a particular focus on improving the evaluation feedback of submitted tasks is the bad and the ugly concern in crowd-based software development.

We are at the beginning of our exploration and the current results represent a first step towards the broader research agenda we are pursuing. In the future, we plan to conduct more extensive analyses to complement the data gathered in Report 1 (description of eligible tasks) and Report 3 (open-ended questionnaire), especially examining other open questions (source 2) applied in the case study focused on further aspects such as: collaboration barriers, suggestions to face them, and the perception about which steps require more interaction support from the crowd, as well as carrying out interviews with some participants in the case study.

We have limitations and topics that remain for future work. As limitations, our study concentrated in one crowdsourcing platform and the strategy of offering to participants of the study only one option of challenge category on Topcoder for task selection led many of them to choose the same kind of task, which was, in this case, First to Finish (F2F). This recommendation also led many of them to carry out, even if in an individual way, the same task. Therefore, we cannot claim the generalizability of the results.

Acknowledgment

The authors thank the students who participated in the study. Alexandre Zanatta thanks University of Passo Fundo for supporting this work.

References

[1] K Mao, Capra Licia, Mark Harman, and Jia Yue, "A Survey of the Use of Crowdsourcing in Software Engineering," *RN*, 2015.

[2] T. D. LaToza and A. van der Hoek, "Crowdsourcing in Software Engineering: Models, Motivations, and Challenges," pp. 74-80, 2016.

[3] K.-J. Stol and B. Fitzgerald, "Two's company, three's a crowd: a case study of crowdsourcing software development," *Proceedings of the 36th International Conference on Software Engineering - ICSE 2014. New York, USA: ACM Press*, p. 187, 2014.

[4] J.D. Herbsleb and R.E. Grinter, "Architectures, coordination, and distance: Conway's law and beyond," *IEEE software, 16(5)*, pp. 63-70, 1999.

[5] D Lakhani , D Garvin, and E Lonstein, "TopCoder (a): developing software through crowdsourcing," *Harvard Bussines School*, 2010.

[6] TopCoder. [Online]. http://www.topcoder.com/

[7] K Mao, Y Yang, Q Wang, Y Jia, and M Harman, "Developer Recommendation for Crowdsourced Software Development Tasks"," *Service-Oriented System Engineering (SOSE), IEEE Symposium,* pp. 347-356, 2015.

[8] W Li, W. T Tsai, and W Wu, "Crowdsourcing for Large-Scale Software Development Crowdsourcing," *Springer Berlin Heidelberg*, pp. 3-23, 2015.

[9] L. Bardin, *Analysis of content*. Lisbon, Portugal, 1977.

[10] K. Boudreau, P. Gaule, K. R. Lakhani, C. Riedl, and A. W Woolley, "From Crowds to Collaborators: Initiating Effort & Catalyzing Interactions Among Online Creative Workers," *Carnegie Mellon University*, 2014.

[11] L. Machado, J. Kroll, S. Marczak, and R. Prikladnicki, "Breaking Collaboration Barriers through Communication Practices in Software Crowdsourcing," *IEEE Conference on Global Software Engineering*, pp. 44-48, August 2016.

[12] A. L. Zanatta, I. Steinmacher, L. Machado, C. R.B de Souza, and R. Prikladnicki, "Barriers Faced by Newcomers in Software Crowdsourcing Projects," *IEEE Software*, p. in press, 2017.

[13] A. Xu, H Rao, Dow, S. P , and B. P. Bailey, "A classroom study of using crowd feedback in the iterative design process," P*roc of the Conference on Computer Supported Cooperative Work & Social Computing*, pp. 1637-1648, February 2015.