# Competence, Collaboration, and Time Management: Barriers and Recommendations for Crowdworkers

Alexandre Lazaretti Zanatta
Computer Science School - PUCRS
Porto Alegre, RS - Brazil
University of Passo Fundo - UPF
Computer Science – ICEG
alexandre.zanatta@acad.pucrs.br

Leticia Machado
Computer Science School
Pontifical Catholic University of Rio Grande do Sul, PUCRS
Porto Alegre, Brazil
leticia.machado.001@acad.pucrs.br

Igor Steinmacher
Department of Computing - UTFPR
Campo Mourão, PR– Brazil
School of Informatics, Computing and Cyber Systems - NAU, AZ– USA
igorfs@uftpr.edu.br

## ABSTRACT

Software crowdsourcing development requires a continuous influx of crowdworkers for their continuity. Crowdworkers should be encouraged to play an important role in the online communities by being active members, but they face difficulties when attempting to participate. For this reason, in this paper, we investigated the difficulties that crowdworkers face in crowdsourcing software development platforms. To achieve this, we conducted a study relying on multiple data sources and research methods including literature review, peer review, field study, and procedures of grounded theory. We observed that crowdworkers face many barriers – related to competence, collaboration, and time management – when making their contributions in software crowdsourcing development, which can result in dropouts. The main contributions of this paper are: a) empirical identification of barriers faced by crowdsourcing software development crowdworkers, and b) recommendations on how to minimize the barriers.

## CCS CONCEPTS

- CCS→Human-centered computing→ Collaborative and social computing→**Empirical studies in collaborative and social computing**

## KEYWORDS

Software crowdsourcing, barriers, crowd workers.

## 1 INTRODUCTION

Crowdsourcing is an approach that involves delegating tasks or micro-tasks to a crowd, an unknown work-force and rises as an option providing access to a scalable workforce of on-line experts [1]. It also promises cost savings, time to market, productivity and flexibility, tapping the intellect of the crowd are evidenced. Recently, crowdsourcing has been adopted in a variety of domains, including software development.



CSI-SE'18, May 27, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5733-3/18/05…$15.00
https://doi.org/10.1145/3195863.3195869

This new approach, known as crowdsourcing in software development (CSD), uses a large pool of potential developers on demand to outsource parts or the entire software project to a crowd of developers [2].

In software crowdsourcing projects, crowdworkers deliver tasks related to different software development steps, including requirements, design, coding, testing, evolution, and documentation [2], [3], [4]. Crowdworkers are one of the key actors in a crowdsourcing model and can bring competitive advantages for the clients who are seeking increased speed of software development and creative solutions [5]. This means that it is vital to motivate, engage and retain crowdworkers to software crowdsourcing projects in order to promote a sustainable community.

LaToza & van der Hoek [6] report that: "in communities that engage in competitions, established experts sometimes win the majority of competitions, discouraging novices and effectively shutting them out of such competitions."

Competition-Based Crowdsourcing Software Development (CBCSD) is a model in which a client requests tasks and pays for its completion. Projects are decomposed into a series of competitions dividing each into tasks. Contestants each provide a competing solution; from these, the platform selects a winning entry and the corresponding workers are paid [6]. This means that the competition model, the crowdworkers compete against each other and this aspect may bring negative impact in order to communicate, share their insights and experiences; and manage the time to be the first to send the best solution. It is worth noting that many problems in crowdsourcing software projects are increased when compared to the traditional software development, as reported by previous research [7], [8], [9] and [10].

According to Howe [1], crowdsourcing is the application of Open Source Software (OSS) principles. In the context of OSS, a study from Steinmacher [11] reported that newcomers usually face many barriers to make their first contribution to a project. We believe that crowdworkers, specifically newcomers to CSD, may face similar difficulties as in OSS development, including technical problems, lack or poor documentation provided, community receptivity issues, tasks with unclear descriptions, cultural differences, and others. In this way, engage and maintain an unknown and specialized group to accomplish tasks in an extremely distributed software development environment is not a trivial task.

In this context, we aim to answer the following research question: "How to support crowdworkers to participate in activities in Competition-Based Crowdsourcing Software Development (CBCSD)?". To achieve it, we conducted an exploratory case study

in Topcoder, one of the biggest CBCSD existing platforms [12] in terms of the number of members, to uncover the barriers faced by crowdworker. After identifying and classifying the barriers, we also analyzed the existing literature to suggest a set of recommendations on how to overcome such barriers.

The remainder of this article is structured as follows. Sec. 2 describes background. Sec. 3 defines the research method. Sec. 4 reports the results. Sec. 5 presents our recommendations to support crowdworkers. Sec. 6 summarizes and we conclude with some suggestions for future work.

## 2 RELATED WORK

The CSD literature is recent and offers a perspective of its elements: the client, the platform, the crowd and the task. From a crowd perspective [13], [14], [15], [16], [17] dedicate some attention to the motivation, the retention and, to the quality of the task produced. [18].

A collaborative model, called Turkomatic, was presented by [18] to aid in the development of complex tasks. They report that when the crowd isn't overlooked or supervised the tasks are executed partially, with questionable quality. However, when they are overlooked, the quality of the task increases significantly.

It was in the work of Steinmacher [11] that authors identified and understood the barriers that the newcomers faced after performing the first contribution to free software development projects. Fifty-eight barriers were identified and grouped into six categories, amongst which: cultural differences, characteristics of the newcomers, reception problems, newcomer's orientation, technical obstacles, and problems with documentation. Based on that realization a portal named FLOSScoach (http://www.flosscoach.com) was developed to support newcomers to their first contribution.

It is possible to conclude, then, that besides being a great opportunity, studies were not identified in the literature which would approach specifically the theme of the barriers faced by crowdworkers in CBCSD.

## 3 RESEARCH SETTINGS

This research is qualitative and is related to understanding some aspect of social life [19]. Qualitative research enables the researcher to choose a collection of data collection techniques to develop the study [19]. We chose to use as multiple research methods: literature review, peer review, field study, and procedures of grounded theory (GT) [20]. Fig. 1 summarizes the research design and the selected methods.

In Phase 1, we conducted a review of the theoretical basis, which involved the review of the literature about crowdsourcing and software crowdsourcing. To identify the barriers that hinder newcomer's contribution in CBCSD we conducted a study relying on two different sources: (a) data gathered from Topcoder platform and (b) feedback from eight newcomers after they attempted to contribute to software crowdsourcing tasks. In (b), the newcomers were asked to select a task of their choice on Topcoder (Development challenge category) and submit their solutions. By collecting and analyzing their feedback, we identified some barriers for newcomers and suggested solutions to overcome such barriers [21]. This phase was performed between March and August 2016.

In the second phase, we conducted a new field study to identify what are the barriers faced by crowdworkers (not necessarily newcomers) while attempting to contribute in Software Crowdsourcing, and also to identify potential solutions to overcome such barriers [22]. Phase 2 complements the qualitative study of Phase 1. This phase was performed between September and December 2016.

In Phase 3, we conducted a field study to assess the barriers identified. Phase 3 complements the qualitative study of Phase 1 and Phase 2. This phase was performed between August and December 2017.

In all phases, all participants voluntarily agreed to join the study.

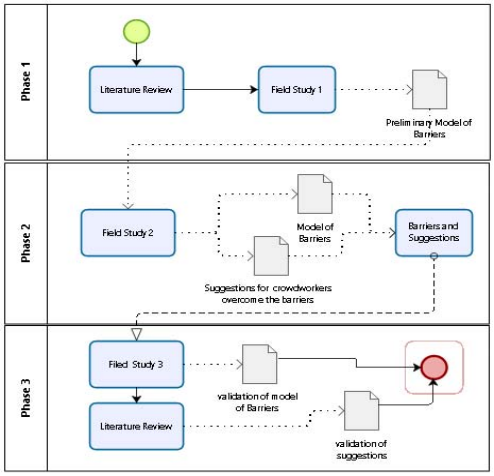



**Figure 1. Research design and the selected methods.**

## 4 RESULTS

In this section, we bring our results summarized per phase.

### 4.1 Phase 1

In Phase 1, we conducted a literature review of the available data repository provided by [3]. We observed a lack of studies faced by newcomers in CBCSD model. In addition, to collect the first data source, we implemented a crawler (using Topcoder public API) and gathered 3,403 Topcoder members who participated in an "Algorithm Competitions". We chose Algorithm Competition because it is the most popular competition in Topcoder. "Algorithm Competitions" are competitions in which all participants compete online and are given the same problems to solve under the same time constraints.

We used the following search criteria, to enable us analyzing newcomers: (a) users' algorithm rating would be at most 1,199, and would be on "green". (b) maximum time since last algorithm competition would be 180 days, i.e., members who have participated in the last 6 months; and (c) user was active on Topcoder during our period of collection (May 23, to Jun 12, 2016). By analyzing the collected data, we found that most of the selected participants had not ever won an algorithm competition. This is an indication that members face barriers to make a successful contribution (i.e., never

made a submission that won a competition) to a software crowdsourcing task.

Our second data source was composed of the feedback of newcomers to crowdsourcing. To gather their feedback, we conducted a post-assignment interview, and qualitatively analyzed the collected data. The goal was uncovering potential barriers faced during the contribution process. During the interview, the participants were asked the following question, inspired by a previous study [11]: "Based on this experience, what are the main barriers faced by newcomers when they want to start contributing to Topcoder?". We invited 8 people who were newcomers to CSD and crowdsourcing platforms. The group was composed of 3 senior Computer Science students attending to a Software Engineering course at University of Passo Fundo (UPF) – Brazil, and 5 developers from software industry with at least 2 years of experience. As a result, we identified some barriers emerged from this analysis [21]. These barriers are: lack of documentation, poor task management, problems understanding code structure or architecture, information overload, poor platform usability, and lack English skills.

## 4.2 Phase 2

In Phase 2, we conducted a field study in the context of a graduate course project on Collaborative Software Development at Pontifical Catholic University of Rio Grande do Sul (PUCRS), Brazil. The group was composed of 21 students attending this course, 13 (62%) with at least 6 years of software industry experience; 16 (76%) were new to CSD.

We selected Topcoder again as the platform for the project to run the study in. The students were introduced to the platform and given some weeks to select a task on Topcoder (Development Challenge Category) to work on. The students were told to work independently and that they had 6 weeks to conclude the task and submit it in the platform. They were asked to observe and respond to any feedback, if given, from the platform. At the end of the assignment, a focus group was conducted to collect the students.

The assignment was associated to the following activities: a) Report 1 – Open experience report: 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 CSD experience in terms of barriers faced while attempting to send their code solution on Topcoder. b) Report 2 – Open-ended online questionnaire: this questionnaire provided us with profiling information on the students' background and Likert scale questions asking about their opinion on the course project.

The students were asked the following question: "What 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 all the answers (Phase 1 and Phase 2) and some barriers emerged from this analysis [22]. We found 20 barriers grouped into five categories: time management, tasks problems, other problems, collaboration, and cognitive problems. Fig. 2 summarizes the barriers identified.

In all phases, for the coding process, we used inductive logic to construct analytical codes. We had inferred theoretical categories from the data and their properties and we wrote memos (notes and diagrams) to create categories as they emerge. The constant comparison between data, memos, categories, and codes was performed. We executed our analysis using the Nvivo® tool software several times to refine the emerging categories and codes. Therefore, using these GT practices [20], we believe that we are developing a cohesive model to represent our phenomenon.

Considering the goal of this article the barriers model was analyzed and we choose only barriers faced in a crowdworkers perspectives. Was used peer-review to map the identified barriers into the four types of crowdsourcing elements: the client, the platform, the crowd and the task.

So, we have excluded "other problems" and "tasks problems" categories and some barriers related only to Client, Platform and, Tasks, keeping just the categories and barriers faced from the crowdworker perspective.

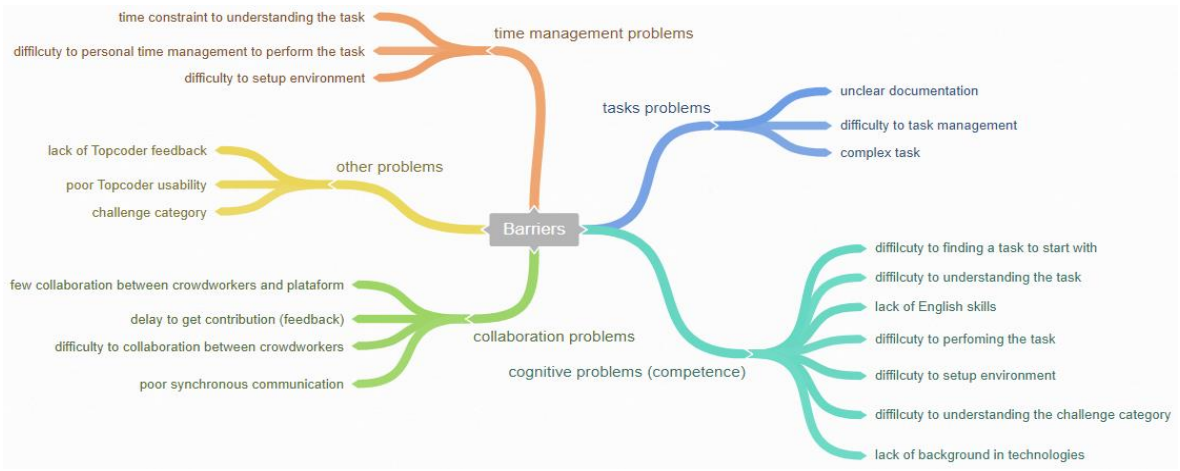



**Figure 2. Barriers model.**

As a result, we could identify three categories and eight barriers as follows.

With regards to the Competence category, crowdsourcing is characterized as well by the capacity to attract a big number of people with competence to perform specific tasks. That is, competence is a characteristic of the crowd which helps the solution of problems, innovation, co-creation, among others [5]. Competence an essential feature of the crowd. According to Hosseini et al. [23], the competence of the crowdworker in performing a complex task is considered core and will be more productive in user-driven crowdsourcing activities. We could identify five barriers within the Competence category, as follows:

Barrier 1 - lack of English skills.

Barrier 2 - difficulty to find a task according to the abilities or profile.

Barrier 3 - difficulty to understand the task.

Barrier 4 - difficulty to setup environment to perform the task.

Barrier 5 - difficulty to perform the task.

For the Collaboration category, first, it is important to state that to collaborate is the action when two or more people work together to reach the same goal,[1] and, in order to do that, people need to share information, communicate, and help each other mutually. The software development activity is essentially collaborative. According to Howe [1] and, MacLean et al. [24] crowdsourcing is a "massive form of parallel collaboration". The communication between members of the project plays an important role in any well-succeeded software project [25]. We could identify two barriers within the Collaboration category, as follows:

Barrier 6 - Lack of English skills to collaborate.

Barrier 7 - Lack of collaboration skills.

The Time management category refers to the capacity of the worker to include some necessary processes to manage the on-time conclusion of the task offered on the platform. That is, the difficulty of the worker in managing time includes processes necessary to understand, to setup environment, to perform and mainly finish the task. One of the characteristics of crowdsourcing is the agility in the resolution of a task, however, this does not guarantee that the task is going to be concluded within the set time, since crowdworkers with knowledge and competence are not always available when needed. It is important to guarantee that enough time was attributed to the task's executioners and that planning is made based on the size and extension of the task. Long and time-taking projects may result in a lack of interest from the crowd to perform the task, causing a smaller number of submissions. We could identify three barriers, "difficulty to time management to learn how to do the task", "time management constraints to execute the task" and "difficulty to time management to configure the environment so to do the task", but we reorganized just one barrier within the Time Management category.

Barrier 8: Difficulty to manage personal time.

## 4.3 Phase 3

In order to assess the barriers identified, we conducted a new field study with post-graduate students (Masters and Doctorate programs) as part of a course project on Collaborative Software Development at PUCRS. The new group was composed of 21 Brazilian students attending to this course, 18 (85%) with at least 6 years of software industry experience; 19 (90%) were new to CSD. We used the Topcoder platform again and repeated the instructions given in phase 2 to new students. The students were asked the following question: "How do you perceive the following barriers about your experience on using Topcoder platform?". We wrote the survey sentences to represent the barriers and reduce potential response bias (Table 1).

Table 1 list the frequency of responses per barriers in a crowdworker perspective, as well as how mode values are distributed. The answers are in a 5-point Likert-scale from Strongly Disagree (1) to Strongly Agree (5) with a Neutral option (3).

**Table 1: Frequency of responses**

| Barriers | Mode | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 8 - It was hard to manage my time available to execute the task. | 5 | 3 | 1 | 0 | 5 | 12 |
| 2 - It was hard for me to find a task and execute it according to my profile or abilities. | 5 | 1 | 3 | 1 | 6 | 10 |
| 4 - I had trouble to configure the necessary environment to perform the task. | 5 | 4 | 2 | 6 | 2 | 7 |
| 5 - Even finding a task according to my profile or abilities, I had difficulty to execute it. | 4 | 2 | 5 | 2 | 9 | 3 |
| 7 - I had difficulties in collaborating with the other members of the platform to perform the task. | 3 | 7 | 0 | 14 | 0 | 0 |
| 3 - Even finding a task according to my profile or abilities, it was difficult for me to understand the task. | 2 | 6 | 8 | 1 | 3 | 3 |
| 6 - When performing the task, I avoided collaborating with other members of the platform because they only communicate in English. | 1 | 11 | 0 | 9 | 0 | 1 |
| 1 - As the platform is in English, I had difficulty in understanding the task. | 1 | 15 | 3 | 1 | 2 | 0 |

**Legend**: 1- Strongly disagree, 2-Disagree, 3-Neutral, 4-Agree, 5-Strongly Agree

In Table 1, we presented ours results ordered by mode. The results show that the barrier that hindered most of our participants while conducting their tasks was the difficulty to manage personal time (barrier 8), with 80% of agreement. This occurs because the groups of students that took part in the study, just like many in the crowd of Topcoder, do not have an exclusive dedication to the platform and use the CSD model in order to obtain extra money.

It is also worth noting that another important barrier, as per 80% of the participants (60% strongly agreeing), was the difficulty to find a task according to the abilities or profile of the crowdworker (barrier 2). This may occur because it is necessary for the crowdworker to be proficient in a certain technology, which may exclude some participants. Maybe a categorization of the challenges by subject could attract more participants and give them a better understanding of the context domain of the challenge. Difficulty to setup environment to perform the task (barrier 4), was also reported as one of the major problems faced by the crowdworkers (56% of

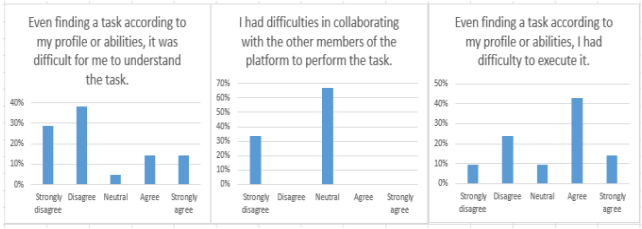
[1] http://dictionary.cambridge.org/dictionary/english/collaboration

agreement). However, in this case, it depends on the task to be performed, which can require or not the complete setup of a local workspace (downloading, building and running the system locally). The results of agreement level of barriers 8, 2 and, 4 may be observed in Fig. 3.


Barrier 8 Barrier 2 Barrier 4

**Figure 3. Results of the Barriers 8, 2 and, 4**

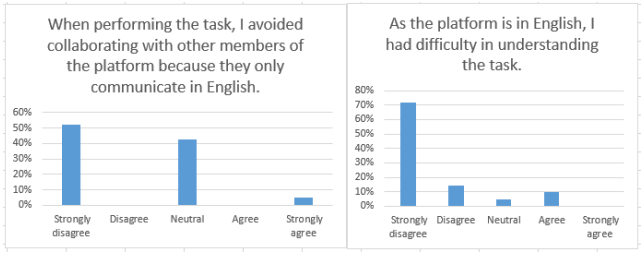
The crowdworker had difficulty to understand the task and execute it, (barriers 5 and 3) even choosing the task according to your profile. This occurs due to the nature of the CSD. The crowdworker faces work fragmentation and scarce context as the tasks are self-contained and intentionally suppress specific business rules from clients who demand the tasks. The results of agreement level of barriers 5, 7 and, 3 may be observed in Fig. 4.




Barrier 5 Barrier 7 Barrier 3

**Figure 4. Results of the barriers 5, 7 and, 3**

Even considering that the study was performed with Brazilian participants, the barrier of the domain of English (barrier 6) was highlighted by only 5% as a problem for interacting with the other participants of the task or of the platform. The results of agreement level of barriers 6 and, 1 may be observed on Fig. 5.




Barrier 6 Barrier 1

**Figure 5. Results of the barriers 6 and, 1**

Regarding those participants who reported strong disagreement to avoid collaborating with other influenced by English idiom, it is possible to consider that participants have collaborated even in English idiom via forum channel (post messages or simply read and gather information from the posts of other task participants). For the participants who answered "neutral", we can interpret who they didn't feel comfortable to communicate via text message using a non-native idiom as shown in Fig. 5 (Barrier 6).

In Fig. 5 (Barrier 1), we can observe that 70% of the participants did not find it difficult to understand the task specifications and requirements written in English. This is interesting because it contrasts with our previous results from Phases 1 and 2, in which participants reported idiom/language as a collaboration barrier. Therefore, it would be interesting to further explore this aspect, to better understand the factors that contribute to the understanding of the tasks specifications or requirements.

## 5 Recommendations to support crowdworkers in CBCSD

The validation is a necessary criterion in the GT so to consolidate the research and show its scientific accuracy. As show Strauss & Corbin [20], the objective of the validation is not to test the model, as happens in the quantitative research, but to compare the concepts and their relationships with the data, and determine if they are in consonance or are appropriate to the investigation.

Based on the identified barriers, literature review and, crowdworkers suggestions, we list some recommendations for Topcoder participants as potential solutions to overcome such barriers. The six recommendations are listed (see Table 2), ordered from most to least frequent barrier.

**Table 2: Recommendations to support crowdworkers**

| # | Recommendations | LR | C | Barrier |
|---|---|---|---|---|
| 1. | Planning and controlling the personal time available is fundamental to execute the task. | X | - | 8 |
| 2. | Making use of a system that helps matching tasks requirements crowdworker profile. | X | X | 2, 3 |
| 3. | Decompose a task into many other tasks (micro-tasks) helps the comprehension of that task. | X | - | 5 |
| 4. | Using a container or virtual machine helps the preparation of the environment to perform the task. | X | X | 4 |
| 5. | Using communication channels like chats or forums is important to the collaboration to understanding and performing the task. | X | X | 6, 7 |
| 6. | Using auto-translation mechanisms (machine translation) ease the comprehension of the task and collaboration | X | - | 1 |

**Legend**: LR – Recommendation from Literature Review
C – Recommendation from participants of case studies.
B – Barriers the recommendation helps overcoming

The recommendations are presented as follows.

### 5.1 Planning and controlling the personal time available is fundamental to execute the task

The Project Management Body of Knowledge (PMBOK) guide, [26] is a set of practices in Project management which, on its 5th edition, considers fundamental to estimate the human resources, equipment, supplies and amount of required material for performing each activity, meaning, to setup environment is something already known and related to project management.

Crowdworker needs to choose a task, when applicable, where the platform indicates an environment for the development of the task as well as, when possible, deliver the possibility of performing the task through other development tools, using, for example, collaborative repositories, in order to maximize your performance.

Additionally, some of the best practices are also recommended like "Use common software without requiring admin installations!"

[27]. The crowdworker needs to evaluate and manage the time to learn how to do the task, also called learning curve. The learning curve is a representation of the average cognitive level of learning for a certain activity [28]. Park et al. [28] mentioned that users spend a significant amount of time learning about the project before effectively taking part in its activities. Another discussion pointed out by Schmid-Druner [29] regards the work of crowds as "free time activities." The authors comment that the workers of European platforms don't intend to perform these activities as their main source of income, but consider it a free time activity parallel to another job. A study from the International Labor Office indicates that for "38% of the American workers and 49% of the Indian workers of AMT, the work constitutes their main source of income". Furthermore, the Topcoder platform has systematically offered educational programs like the Topcoder Education Week Marathon Match to their users as a way to support participation in competitions.

In order to estimate the time that will be allocated to execute the task, it is necessary to plan and manage a timeline, define some activities, estimate resources and duration of activities and, finally, control the defined timeline [26]. Hoßfeld et al. [27] mention that the decision to work on is affected by the estimated time that was assigned by the platform before executing a task. They indicate that this is the reason why, for the platform, it is fundamental to be precise in estimating the execution time of the task and, for the worker, to have mechanisms that help him define the execution time of the task.

Therefore, crowdworkers need previous technical knowledge (cognitive effort), as well as efficient ways to manage their time in order to successfully deliver a task.

## 5.2 Making use of a system that helps matching tasks requirements crowdworker profile

We hypothesize that the barrier "difficulty in finding a task" can be caused by usability issues of the platform. As stated by De Souza et al. **[**30**]**, a virtual community system with "good" usability must have a browsing structure which allows the members to easily browse and find what they need. Yang et al. [31] investigated worker decision factors providing better decision support in order to improve the success and efficiency of CSD. The authors proposed a dynamic crowd worker decision support problem ("DCW- DS") and an analytics-based decision support methodology to guide workers in acceptance of offered development tasks from Topcoder platform. Moreover, searching for tasks that only maximize the expected earnings is pointed by Horton et al. [32] as an important motivation in the process of finding a task.

Therefore, crowdworkers should have access to some recommending system to support find tasks according their profile.

## 5.3 Decompose a task into many other tasks (micro-tasks) helps the comprehension of that task

Tajedin et al. **[33]** suggest that projects that can be broken into small modules with clear requisites and limited interdependencies are keener to have success. However, when breaking a task in software development, it is necessary to establish a balance between giving a sufficiently detailed specification so that the task may be executed, and the time required to specify and to perform the task itself. LaToza et al. [34] proposed a platform ("CrowCode"), for sharing knowledge through a question and answers system. The authors state that this platform, based on micro-tasks, aids in the understanding of the tasks by the crowdworker. Dustdar et al. [35] present a tool that helps the project manager to delegate the tasks to the crowd according to their profile, which can contribute to the understanding of the task by the worker. Additionally, Preist et al. [36] suggest that the payment based on competition "works well" for qualified workers.

Other studies [37], [38], [39], [40] present the challenges of crowdsourcing software structured in six areas, among them, the breaking of tasks. The work broken in a set of significantly smaller tasks (micro-tasks) is a key question when defining an outsourcing scenario. Software development tasks using crowdsourcing usually depend on each other. This becomes a problem considering that different developers perform their tasks without knowing how their changes are going to be integrated into the final product, which is the result of the sum of all submissions. When dealing with software development and the fact that tasks are more complex and dependent of each other, the challenge is to find an adequate breaking of the product in tasks which can be sent to the crowd. According to Stol et al. [5] to break the project into small modules, to have "clear" requisites and to limit the dependency of tasks can define the success of the project.

For the barrier "difficulty to perform the task", the capacity is substantially related to the conclusion of the task, besides the fact that only workers able to perform the task, often finish them. Some studies [10], [41], add that in complex environments it is fundamental to consider the experience of the worker in delegating the tasks. They also state that the lack of knowledge is a big source of misunderstandings in the traditional software development and is increased in the crowdsourcing software. Therefore, crowdworkers should decompose a task into many other tasks.

## 5.4 Using a container or virtual machine helps the preparation of the environment to perform the task

The barrier "Difficulty to setup environment to perform the task" or prepare a computational structure (setup environment)", with specific software and hardware, among other structures, was pointed by the participants as one of the main barriers for the execution of some of the tasks in crowdsourcing in software projects. This is a barrier that is difficult to overcome, since it depends on information from the client, the platform, the documentation associated with the platform, as well as from other workers in the platform. To offer a pre-configured computational structure, so that the newcomers in OSS environments could contribute for the first time, was a suggestion discussed by [42].

In this context, there are some actions that aid in the environment preparation process for the execution of tasks. WikiIDE, for example, offers a collaborative development environment with support to specific services and applications which may help in the development of tasks in crowdsourcing software projects. The use of virtual machines with integrated development environments (IDEs) is also an option. Another possible solution is the use of IDEs under the concept of cloud computing. Some of the examples of these IDEs are Codeanywhere, Cronapp and Onion Cloud.

Open platforms for developers like Docker may help the preparation of setup environments for the execution of tasks in CSD because, according to Docker platform "A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings." Docker uses the concept of "containers", which aims to facilitate the creation and administration of isolated environments for the development of computational solutions. Other examples of container management tools include http://portainer.io/ and http://rancher.com/.

Virtual machines integrated with the IDEs or with containers, are possibilities for the crowdworkers to prepare their environment for the execution of a certain task which, however, can be considered of difficult solution depending obviously on the type of task to be understood, planned and executed. The constant demand and diversity of types of task (micro and macro), their classification (low, moderate or high complexity), limitations of the platform, low cognitive demand of the worker all tied to the technological evolution are impeditive factors for the assembly of an environment which may serve for the execution of any task.

Therefore, the crowdworker needs to have enough competence to prepare to environment to perform the task.

### 5.5 Using communication channels like chats or forums is important to understanding and performing the task

The barrier "lack of English skills" to express or understand a non-native language affects negatively the collaboration among crowd members while performing their tasks [43], [44]. The participants of our study stated that there was little collaboration among the members before, during and after the execution of the task. As seen, the barrier "lack of collaboration skills" is essential in software development activities, and in crowdsourcing software, this need for interaction generally increases [7], [8], [9], [45]. Barriers related to communication, cultural differences and different time-zones decrease the collaboration among crowd members [10].

Saxton et al. [46] show that the level of collaboration can vary substantially depending on the crowdsourcing model, and the complexity of the task adopted for the project. Thus, it is important to provide communication mechanisms to the crowd members, such as chat rooms and feedback channels, so that the members can interact while contributing [47]. The study participants indicated that the use of chat tools potentially increments collaboration. Hoßfeld et al. [27] also evidenced that the use of feedback channels can enhance communication among workers, in crowdsourcing settings.

Kittur et al. [44] add and understand that it is essential to keep the worker motivated, offering him constant feedback of the tasks besides using an efficient payment system – when that's the case, provide better interaction among the workers in the crowd. It is worth noting that "Topcoder is able to implement team-sourcing where crowds can collaborate in multiple ways to complete complex projects" [48].

As in any distributed software development setting, people need efficient relationship mechanisms to create a shared vision through interactions. Then, it is fundamental that the stakeholders of CSD projects play a facilitator role in this process, offering opportunities to discuss with and receive opinions from the crowdworker.

### 5.6 Using auto-translation mechanisms (machine translation) ease the comprehension of the task and collaboration

Lack of English skills is still a problem and an important barrier in the development of tasks on CBCSD. Some studies [49], [50], [51], [52], [53], present solutions for the simultaneous translation between many languages through techniques like "Machine translation" (MT) and the computer-assisted language learning. These solutions may help the worker to perform tasks in crowdsourcing software projects. The lack of specialists in translation represents a problem for the different and growing markets in all the world, as indicated by Taravella et al. [52]. One of the solutions proposed for the lack of human resources is the usage of automated translation tools. Therefore, the crowdworker needs to have enough competence in the English language to express or understand a non-native language to understand tasks and collaboration.

## 6 CONCLUSIONS

In summary, we identified a set of barriers that hinder crowdworker contributions; and suggested a small set of recommendations to overcome these barriers. As any other empirical work, we have limitations and topics that remain for future work. As one limitation, our study concentrated on the most used competition-based crowdsourcing platform, and we cannot say that the behavior identified here will replicate in other platforms. We executed a qualitative study using interviews, which, potentially, has various limitations. In future work, we will apply other research methods to help us understanding the barriers in crowdsourcing software projects. Identifying suitable participants is always a difficult balance. In this case, software engineering students have been used, and they are very likely to be crowdworkers for software projects, but there is such a wide variety of profiles for potential crowdworkers that this will always be a limitation. Moreover, we intend to work with different types of challenges, platforms, and more participants to continue to identifying patterns, behaviors, and ways to improve the way crowdworkers contribute to CSD. Therefore, we conclude that the crowdworkers need competency and an efficient time management effort to take part collaboratively in tasks of the CBCSD of the Topcoder platform.

## ACKNOWLEDGMENTS

The authors thank professor Rafael Prikladnicki for his insightful comments on drafts of this paper, the students and software engineering professionals who participated in the studies, professor Sabrina Marczak, Master Student Luis Vaz and thanks University of Passo Fundo and PUCRS for supporting this work. This project is partially funded by FAPERGS (project 17/2551-0001/205-4), CNPq (Grant #430642/2016-4); and FAPESP (Grant #2015/24527-3).

## REFERENCES

[1] J. Howe, "The rise of crowdsourcing," *Wired magazine, vol. 14,* 2006.

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

[3] K. Mao, C. Licia, M. Harman and J. Yue, "A Survey of the Use of Crowdsourcing in Software Engineering," *RN,* 2015.

[4] W. Wu, W. Li and W. T. Tsa, "An evaluation framework for software crowdsourcing," *Frontiers of Computer Science 7(5),* pp. 694-709, 2013.

[5] K.-J. Stol, B. Caglayan and B. Fitzgerald, "Competition-Based Crowdsourcing Software Development: A Multi-Method Study from a Customer Perspective," *IEEE Transactions on Software Engineering,* 2017.

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

[7] M. N. Wexler, "Reconfiguring the sociology of the crowd: exploring crowdsourcing.," *International Journal of Sociology and Social Policy, v. 31, n. 1/2,* pp. 6-20, 2011.

[8] B. B. Bederson and A. J. Quinn, "Web workers unite! addressing challenges of online laborers.," *CHI'11 Extended Abstracts on Human Factors in Computing Systems. ACM,* pp. 97-106, 2011.

[9] W. Wu, W. T. Tsai and W. Li, "Creative software crowdsourcing: from components and algorithm development to project concept formations.," in *International Journal of Creative Computing, 1(1)*, 2013.

[10] B. Satzger, R. Zabolotnyi, S. Dustdar, S. Wild, M. Gaedke, S. Göbel and T. Nestler, "Toward Collaborative Software Engineering Leveraging the Crowd.," *Economics-Driven Software Architecture,* pp. 159-181, 2014.

[11] I. F. Steinmacher, "Supporting newcomers to overcome the barriers to contribute to open source software projects," *Doctoral dissertation, Universidade de São Paulo,* 2015.

[12] Topcoder, "Topcoder website," [Online]. Available: http://www.topcoder.com.

[13] R. Jayakanthan and D. Sundararajan, "Enterprise crowdsourcing solutions for software development and ideation.," *Proceedings of the 2nd international workshop on Ubiquitous crowdsouring. ACM,* pp. 25-28, 2011.

[14] R. Jayakanthan and D. Sundararajan, "Enterprise crowdsourcing solution for software development in an outsourcing organization.," *International Conf. on Web Engineering. Springer Berlin Heidelberg,* pp. 177-180, 2011.

[15] 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.

[16] L. R. Varsshney, "Participation in crowd systems.," *In: Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on IEEE,* pp. 996-1001, 2012.

[17] M. Vukovic, "Crowdsourcing for Enterprises," *2009 World Conference on Services - I. Los Angeles, CA: IEEE,* p. 686–692, 2009.

[18] P. G. Ipeirotis, "Analyzing the amazon mechanical turk marketplace. XRDS: Crossroads,," *The ACM Magazine for Students,,* pp. 16-21, 2010.

[19] D. Yvonne, J. Michael and S. Janice, "Editorial: For the Special Issue on Qualitative Software Engineering Research.," *Information and Software Technology. 49 (6),* p. 531–539, Jun 2007.

[20] A. Strauss and J. Corbin, Basics of qualitative research: Techniques and procedures for developing grounded theory, 2ª ed., Sage Publications, 1998.

[21] A. L. Zanatta, I. Steinmacher, L. Machado, C. R. d. Souza and R. Prikladnicki, "Barriers Faced by Newcomers in Software Crowdsourcing Projects," *IEEE Software,* 2017.

[22] L. Machado, A. Zanatta, S. Marczak and R. Prikladnicki, "The good, the bad and the ugly: an onboard journey in software crowdsourcing competitive model.," *In Proceedings of the 4th International Workshop on CrowdSourcing in Software Engineering*, Buenos Aires, 2017.

[23] M. Hosseini, K. Phalp, J. Taylor and Ali, "The four pillars of crowdsourcing: A reference model," *IEEE Eighth International Conference on Research Challenges in Information Science (RCIS),* pp. 1-12, May 2014.

[24] D. MacLean, K. Yoshida, A. Edwards, L. Crossman, B. Clavijo, M. Clark and M. Caccamo, "Crowdsourcing genomic analyses of ash and ash dieback–power to the people.," in *GigaScience, 2(1)*, 2013.

[25] S. Panichella, G. Bavota, M. D. Penta, G. Canfora and G. Antoniol, "How developers' collaborations identified from different sources tell us about code changes," *ICSME. Conference on Software Maintenance and Evolution,* 2014.

[26] C. S. Snyder, "A Guide to the Project Management Body of Knowledge: PMBOK Guide" 2014.

[27] T. Hoßfeld, M. Hirth, J. Redi, F. Mazza, P. Korshunov, B. Naderi and C. Keimel, "Best Practices and Recommendations for Crowdsourced QoE-Lessons learned from the Qualinet Task Force" Crowdsourcing". 2014.

[28] Y. Park and C. Jensen, "Beyond pretty pictures: Examining the benefits of code visualization for open source newcomers," *5th IEEE International Workshop Visualizing Software for Understanding and Analysis. (VISSOFT),* p. 3, 2009.

[29] M. Schmid-Druner, "The Situation of Workers in the Collaborative Economy," European Parliament, 2016.

[30] C. S. De Souza and J. Preece, "A framework for analyzing and understanding online communities," *Interacting with computers, v. 16, n. 3,* pp. 579-610, 2004.

[31] Y. Yang, M. R. Karim, R. Saremi and G. Ruhe, "Who Should Take This Task?: Dynamic Decision Support for Crowd Workers," *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement,* p. 8, September 2016.

[32] J. J. Horton and L. B. Chilton, "The labor economics of paid crowdsourcing.," *Proc. of the 11th ACM conference on Electronic commerce,* pp. 209-218, 2010.

[33] H. Tajedin and D. Nevo, "Determinants of success in crowdsourcing software development"," *Proceedings of the 2013 annual conference on Computers and people research. ACM,* pp. 173-178, 2013.

[34] T. D. LaToza, W. B. Towne, C. M. Adriano and A. Van Der Hoek, "Microtask programming: Building software with a crowd.," *27th annual ACM symposium on User interface software and technology. ACM,* pp. 43-54, October 2014.

[35] S. Dustdar and M. Gaedke, "The social routing principle.," *Internet Comput. 15 (4),* pp. 80-83, 2011.

[36] C. Preist, E. Massung and D. Coyle, " Competing or aiming to be average?: Normification as a means of engaging digital volunteers," *Proc. CSCW,* pp. 1222-1233, 2013.

[37] T. D. LaToza, W. B. Towne, A. van der Hoek and J. D. Herbsleb, "Crowd development," *6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). IEEE,* 2013.

[38] K. Mao, Y. Yang, M. Li and M. Harman, "Pricing crowdsourcing-based software development tasks," *Proceedings of the 2013 international conference on Software engineering, IEEE Press,* pp. 1205-1208, 2013.

[39] T. D. LaToza, M. Chen, L. Jiang, M. Zhao and A. Van Der Hoek, "Borrowing from the crowd: A study of recombination in software design competitions.," *Proceedings of the 37th International Conference on Software Engineering-Volume 1. IEEE Press,* p. 551, 2015.

[40] X. Peng, M. Ali Babar and C. Ebert, "Collaborative Software Development Platforms for Crowdsourcing.," *IEEE Software, v. 31, n. 2,,* pp. p. 30–36, 2014.

[41] T. Straub, H. Gimpel, F. Teschner and C. and Weinhardt, "How (not) to Incent Crowd Workers - Payment Schemes and Feedback in Crowdsourcing," *Business & Information Systems Engineering: Vol. 57: Iss. 3,* pp. 167-179, 2015.

[42] C. Hannebauer, M. Book and V. Gruhn, "An exploratory study of contribution barriers experienced by newcomers to open source software projects.," in *Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering. ACM,*, 2014.

[43] C. Codagnone, F. Abadie and F. Biagi, "The Future of Work in the 'Sharing Economy', Market Efficiency and Equitable Opportunities or Unfair," *JRC Science for Policy Report,* 2016.

[44] A. Kittur, J. V. Nickerson, M. S. Bernstein, E. M. Gerber, A. Shaw, J. Zimmerman, M. Lease and J. J. Horton, "The future of crowd work," 2013.

[45] O. Alonso, D. E. Rose and B. Stewart, "Crowdsourcing for relevance evaluation. In: ACM SigIR Forum. ACM," pp. 9-15, 2008.

[46] G. D. Saxton, O. Oh and R. Kishore, "Rules of crowdsourcing: Models, issues, and systems of control.," in *Information Systems Management, v. 30, n. 1,* , 2013.

[47] X. Lu, C. W. Phang and J. Yu, "Encouraging participation in virtual communities through usability and sociability development: An empirical investigation.," *Acm Sigmis Database, v. 42, n. 3,,* pp. 96-114,, 2011.

[48] E. Bari, M. Johnston, W. Wu and W. T. Tsai, "Software Crowdsourcing Practices and Research Directions," *Service-Oriented System Engineering (SOSE), 2016 IEEE Symposium on IEEE,* pp. 372-379, March 2016.

[49] K. J. Dunne, Computer-Assisted Translation. The Encyclopedia of Applied Linguistics, 2013.

[50] A. Trujillo, Translation engines: techniques for machine translation., Springer Science & Business Media, 2012.

[51] L. Bowker and D. Fisher, "Computer-aided translation," *Handbook of translation studies, v. 1,* pp. 60-65, 2010.

[52] A. Taravella and A. O. Villeneuve, "Acknowledging the needs of computer-assisted translation tools users: the human perspective in human-machine translation," *The Journal of Specialised Translation, v. 19,* pp. 62-74, Jan. 2013.

[53] U. Muegge, "Teaching computer-assisted translation in the 21st century, Alles hängt mit allem zusammen: Translatologische Interdependenzen.," *Festschrift für Peter A. Schmitt,* pp. 137-146, 2013.