AALBORG UNIVERSITY
STUDENT REPORT

Study Board of Computer Science
Selma Lagerløfs Vej 300
9220 Aalborg Øst
www.cs.aau.dk

Title:
Crowdsourcing software development: problems experienced from the developers' perspective

Semester:
10th

Semester theme:
Master Thesis

Project period:
Spring semester 2017

ECTS:
30

Supervisor:
John Stouby Persson, PhD
Associate Professor
john@cs.aau.dk

Author:
Svilen Varbanov Kondov
skondo15@student.aau.dk

Date of completion:
September 26th, 2017

Page count:
30

Abstract:

Crowdsourcing software development is an innovative and emerging approach to software development. The current paper builds on previous study conducted by the author which synthesized available literature on inherent benefits and challenges of using crowdsourcing in software development. The purpose of this thesis was to gain deeper understanding of the Topcoder's community of developers, more specifically it was aimed at identifying the most pervasively raised problems related to participating in crowdsourced software development, and uncovering how Topcoder's community management is addressing these problems.

Crowdsourcing software development: problems experienced from the developers' perspective

Svilen Varbanov Kondov

MSc in IT Design and Application development

Department of Computer Science

Aalborg University

skondo15@student.aau.dk

Abstract: Crowdsourcing software development is an innovative and emerging approach to software development. The current paper builds on previous study conducted by the author which synthesized available literature on inherent benefits and challenges of using crowdsourcing in software development. The purpose of this thesis was to gain deeper understanding of the Topcoder's community of developers, more specifically it was aimed at identifying the most pervasively raised problems related to participating in crowdsourced software development, and uncovering how Topcoder's community management is addressing these problems.

Keywords: crowdsourcing; software development; Topcoder; netnography; open innovation; community of practice; pragmatism;

Supervisor: John Stouby Persson, PhD

Associate Professor

Department of Computer Science | Aalborg University

john@cs.aau.dk

1. Introduction

The recent years of technological development, in combination with the rapid spread of the Internet, led to enormous changes in the way a modern company is looking at the world. Many business related areas such as strategy, product development, organizational and human resource management, have in many ways undergone a fundamental paradigm shift.

While companies previously were focused on knowledge accumulation and product development internally which subsequently had to be protected at all costs, today's focus is increasingly gaining access to creativity and expertise from the company's external environment. This is a paradigm shift from closed to open innovation model, where companies should let the innovations freely transfer inward and outward through their boundaries, including outside companies and the consumers themselves (Chesbrough, 2003).

Internet becoming widely accessible has led to that the majority of the developed world is nowadays connected to a global digital community. The rapid advancement in the information and communication technologies, leading to removal of temporal and spatial barriers, means that it is now possible to involve individuals with specialized knowledge located on the other side of the globe in projects that may otherwise have difficulties finding the required expertise. However, while big companies had previously been able to integrate expertise from different countries around the world, nowadays this is achieved by medium-sized companies as well. The modern information and communication technologies create the prerequisites for the modularization of activities and thus facilitate the collaboration with and between external workers. Here is where the term "crowdsourcing" comes into play, contributing to the "democratization" of innovation and knowledge access in the global economy (Bergvall-Kåreborn and Howcroft, 2014).

Crowdsourcing is a portmanteau word for "crowd" and "outsourcing", in simple words meaning outsourcing to a crowd of individuals. It was first coined by the journalist Jeff Howe (2006), who defines it as "*the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals".*

Crowdsourcing is based on the concept of "the wisdom of crowds" (Surowiecki, 2005), which means that the collective wisdom of a large group of individuals most often exceeds the wisdom of the individual person.

Some cases of crowdsourcing are leaning towards the open source idea of collaboration without direct monetary compensation for the participants, such as the websites Wikipedia and StackOverflow. The online encyclopedia Wikipedia is based on the concept of crowdsourcing knowledge, where users contribute with content in their areas of interest and expertise, which, when all knowledge is gathered, creates an always up to date encyclopedia that rarely contains errors as the articles are continuously reviewed by other users who are able to correct and update them. Wikipedia is thus based on the idea of openly shared content, which is maintained by the public without limiting its rights over or access to that content.

Crowdsourcing has been applied to solve problems in various domains such as marketing, idea generation, design and product development, knowledge sharing, etc. In a general sense, the crowdsourcing process is usually facilitated by an online platform, where a community of problem solvers is built around the same interests or practice. When an organization is facing a problem or a challenge, it can then seek a solution to this problem through an open call in the crowdsourcing community, which will in turn produce a number of ideas or solutions to this problem, usually either through collaboration or competition, rewarding the best solution in exchange for the efforts of the crowdsourcing worker.

1.1. Crowdsourcing in software development

Crowdsourcing has been successfully applied also in software development (Lakhani et al., 2010; Stol and Fitzgerald, 2014; Hasteer et al., 2016), and Stol and Fitzgerald (2014) define the concept in this particular context as "*the accomplishment of specified software development tasks on behalf of an organization by a large and typically undefined group of external people with the requisite specialist knowledge through an open call.*"

Currently the most famous platform for crowdsourcing software development is called "Topcoder", a pioneer operating since 2001 that enables enterprises and startups alike to tap into specialized software engineering expertise on demand. Its community of more than 1 million registered members worldwide helps to accelerate innovation and solve its clients' challenging problems in different areas of the software development process.

Crowdsourcing in software development context is still an emergent area of inquiry and it has a relatively scarce application in practice. Nevertheless, the socio-technical character of the phenomenon attracts more and more attention by researchers and allows to be examined from

three different perspectives of the three main actors in the crowdsourcing process – the platform, the client organization, and the worker (Stol and Fitzgerald, 2014).

Previous literature study of extant research on the topic from a customer perspective, showed that as much as there are benefits to be realized, there are even more challenges to be taken into account by the organizations that are considering using crowdsourcing in their software development activities (Kondov, 2017).

Documented benefits were related to cost reduction, higher quality of solutions, faster time-to-market, flexibility and scalability of the workforce, harnessing creativity and diversity of solutions provided. On the other hand, challenges were related to coordination and communication as means for collaboration, project workflow planning and control, ensuring confidentiality and Intellectual Property protection, efficient integration with internal development processes, effective task decomposition and design, assigning and motivating the most suitable workers for the task, cost and time effectiveness, and quality assurance (Kondov, 2017).

The phenomenon of crowdsourcing software development has three inseparable elements – the platform, the customer organizations, and the community of developers. While there are studies conducted focusing on the technical aspect of designing a platform or the organizational perspective of the inherent benefits and challenges of actually using crowdsourcing, not much research has been done taking the developers perspective. The developers' community is an integral part of the Topcoder platform and similar to any other organization, its members are considered a high valuable asset. Therefore this study will take the developers' perspective and will try to understand their community culture and what pervasive problems do they experience when participating.

1.2. Problem formulation

As noted earlier, a software development crowdsourcing platform, such as Topcoder, relies to a big extent on its community of developers to gain competitive advantage over traditional companies offering systems development and IT consultancy services. In order to provide its clients with an opportunity to capture the previously noted inherent benefits of using crowdsourcing in the software development process, Topcoder must aim at not only attracting the most intelligent, innovative and creative minds, but also at keeping them "aboard". This is directly affected by the developers' motivations, aspirations and even more importantly the problems they experience when participating in contests on the platform.

The purpose of this thesis is to gain deeper understanding of the developers' community and culture, more specifically focusing on the problems they experience when it comes to participating in solving challenges and competitions on the Topcoder platform. Furthermore we will explore whether there is adequate response by community management to the raised problems and how are they addressing the latter. The need for such understanding and investigation leads to the following research questions, taking a developers perspective:

RQs:

1. *What are the most pervasively raised problems in the community of Topcoder related to participation in crowdsourced software development?*
2. *How are these problems addressed by Topcoder's community management?*

Answering the research questions would be directly useful to the platform owners and indirectly to the platform clients. Existing problems causing dissatisfaction, demotivation and discontent can lead to reflux of previously active developers, which subsequently leads to decrease in the knowledge base available in the community and lowering the potential for producing high quality innovative solutions for the clients of the platform. Understanding what are the most pervasively raised problems in the community and whether they are addressed properly by the community management would show to Topcoder areas that need change and improvement, which in turn will lead to more satisfied community members and better service offering to the platform clients.

A fairly new method of "virtual" ethnography called "netnography" was decided to be particularly suitable for the purpose of this thesis and for answering the research questions. For a long time it has been arguable whether the results of qualitative research studies are generalizable, as often they are directly compared to quantitative studies that rely on statistical methods and random sampling from the population. According to Morse (1999), such comparison is not appropriate and the criterion for generalizability between qualitative and quantitative studies should be different. In quantitative research, representativeness of the sample is ensured using demographic characteristics of the study population. In qualitative research, each participant in the relatively small sample has been selected purposefully for the contribution he or she can make toward the emerging theory, applicable to all similar situations and problems, regardless of the demographic comparability between the groups (Morse, 1999). Therefore we argue that the current study's findings can be generalized situationally and would have direct implications for other crowdsourcing software development platforms.

2. Theoretical framework

The following chapter describes open innovation as a concept and how it is related to crowdsourcing. Subsequently, the community of practice concept is presented, contributing to understanding the nature of communities and how are they formed. In addition, we look at what is the meaning of "problem" through the lens of pragmatic philosophy.

2.1. Open innovation and crowdsourcing

For most of the 20th century, corporate business models have been characterized by closed innovation, where successful innovation largely depended on internal R&D processes and how well a company was able to protect and control its own intellectual property rights (Chesbrough, 2003). Open innovation has become one of the most discussed topics in innovation literature during the past decade, often relying on empirical studies to focus on finding out how companies can best implement and benefit from open innovation (Dahlander and Gann, 2010). However, being open in the innovation process has many aspects, which affect how a given company should implement open innovation and how the company will benefit from it, as moving from closed to open innovation model is a long way to go for most companies.

If we look back into the innovation literature, Cohen & Levinthal (1990) came up with a concept called "absorptive capacity", which is about a company's ability to absorb and exploit external knowledge in the innovation process. Rothwell (1991) found that successful innovation in a company largely depends on the "networking" capabilities with external actors. Additionally, the not-invented-here (NIH) syndrome, which deals with a company's reluctance towards the use of external resources, has also been a major issue in the earlier innovation literature (Rigby and Zook, 2002). According to Chesbrough (2003), the open innovation logic embraces external ideas and knowledge in conjunction with internal R&D, which offers new ways to create and capture value for the organization. The common denominator of all these concepts relates to the management of an organization's external dimension. Chesbrough (2006) is the first one to name the phenomenon and makes it easier to relate to, defining open innovation as "*the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively. [This paradigm] assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology*".

Crowdsourcing is often associated with open innovation, thus it would be beneficial to examine in more details their perceived relationship. Looking at previous studies, there seem to be no consensus on the matter among authors. Some are treating crowdsourcing and open innovation as synonymous (Chanal-Fasan and Caron, 2008), while others see crowdsourcing as being a sub-type of open innovation (Nambissan and Sawhney, 2007). Hallerstede (2013) examines open innovation platforms, which include crowdsourcing platforms as well, from a socio-technical systems perspective. A socio-technical system comprises a "social" and "technical" subsystems which are influenced by an environment where input is turned into output.

Phillips (2011) proposes a typology of open innovation, where he describes crowdsourcing as a way of implementing open innovation. His view is supported by Osterwalder (2010) who in his book on business model generation, uses a crowdsourcing platform as an illustration of open innovation business model, basically meaning that crowdsourcing and open innovation can use the same business model. Despite the many similarities, Schenk and Guittard (2009) make a clear distinction between the concepts, pointing out that open innovation is exclusively related to innovation processes while crowdsourcing can be applied in many different areas. In summary, we can conclude that there is a shared space between crowdsourcing and open innovation as concepts, however not every open innovation initiative involves crowdsourcing, nor every crowdsourcing activity is facilitating open innovation.

2.2. Communities of practice

The concept of community of practice was introduced by Lave and Wenger (1991) and further developed by Wenger (1998). In the former text, it is defined as "*a system of relationships between people, activities, and the world; developing with time, and in relation to other tangential and overlapping communities of practice*" (Lave and Wenger, 1991). Wenger (1998) himself emphasizes that "*we all belong to communities of practice. At home, at work, at school, in our hobbies – we belong to several communities of practice at any given time. And the communities of practice to which we belong change over the course of our lives."*

Wenger (1998) has identified three dimensions by which practice is associated with creating coherence in the community:

- Mutual Engagement: Practice occurs when people do something together and keep on doing it over time. A community of practice is therefore only possible when one group of people

mutually engage in a particular practice. That does not mean that the community members necessarily are a uniform group characterized by consensus around the common practice. On the contrary, diversity and a continuous negotiation of the common practice is usually one of the characteristics of the community of practice.

- A joint enterprise: The cohesion of the community of practice is achieved through the presence of a joint enterprise, which reflects the collective process of negotiation and the full complexity of mutual engagement. It evolves in the very process of members pursuing it and creates among participants integral relations of mutual accountability, despite all the factors that are beyond their control,
- A shared repertoire: Although the members of the community of practice regularly renegotiate the practices through their mutual engagement, their cohesion comes down to a common repertoire of resources, tools and values. For example, communities of practice are always characterized by an internal jargon with linguistic formulations that members are able to decode and use, but which an outsider will be having difficulty understanding. However, the shared repertoire can also be expressed physically (equipment, attire, etc.) or through processes (practice techniques, styles, etc.).

However, the term "community of practice" has been used with a great deal of variety in several academic fields including organizational studies and information science, particularly the topics of knowledge management and organizational learning. It is used primarily either as a conceptual lens to examine the situated social construction of meaning or as a concept referring to a virtual community or informal group backed up by an organization to facilitate knowledge sharing or learning (Cox, 2005).

In their study Brown and Duguid (1991) put an emphasis that an organization should recognize the value of fostering informal networks which actually work out how to get the job done and generate innovative solutions to the raised problems, in comparison to Lave and Wenger (1991) where the reproduction of existing knowledge is a central theme. In Wenger's later work there is a noticeable shift in his perspective, focusing on the community of practice as valuable management tool facilitating innovation and problem solving in large organizations (Wenger et al., 2002). The community of practice as a concept even becomes redefined as "*groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis*" (Wenger et al., 2002). This definition describes the concept as a different idea compared to Wenger's earlier book (Wenger, 1998). Here the emphasis is on people interested in the same thing, not

necessarily closely tied together in a joint enterprise. According to Cox (2005) the idea implied is "*to create or foster new groupings of people who work on similar or parallel, not joint, enterprises (practices), effectively to invent new practices*", where "*ethically there has been a shift from a concern to reveal and celebrate the value of what people know, especially in seemingly routine or mechanical jobs, to a concern to design a tool for management to manage 'knowledge workers' and experts in blue chip companies."* Furthermore he acknowledges that often such communities are virtual, as inevitably disparate individuals from large multinational organizations are geographically dispersed (Cox, 2005). Such shift might be surprising given that the original meaning of the concept "community of practice" was about face to face interaction. The notion of virtual community is quite ambiguous actually. A virtual or online community is referring to high levels of interactivity and active behavior among web site users, which has to be distinguished from web sites which merely push content to a number of passive users (Cox, 2005). According to von Wartburg et al. (2004), the virtual communities of practice are "*communities of practice characterized by at least partially virtual interactions. They are informal groups of people that share expertise and passion for actual practice within and on behalf of an organization.*" The virtual communities of practice are a desirable form of "virtual communities", as they are an effective organizational instrument for knowledge creation where "learning in practice" takes place as compared to traditional ways of interaction. Members of the community are bound together by encountering common problems in the working process, sharing their expertise and experience, and desire to gain knowledge from each other (von Wartburg et al., 2004).

It is in the sense described by Wenger et. al. (2002) and von Wartburg et al. (2004) that a community of practice as a concept could be viewed as very similar to the crowdsourcing community of Topcoder. In the current study's context, there is an online community created and nurtured around an interest of participation and competition in software development activities, within which there is a specific work practice. Therefore this view relates also to the existing literature where communities of practice have been studied in a workplace management context as they are considered very effective for problem solving. Often the community of practice concept is perceived to be a useful concept in connection with understanding, for example, the efficiency of corporate workflows (Duguid, 2008) or situated learning, which is embedded within an activity, context and culture (Lave and Wenger, 1991). Lave and Wenger's theory on situated learning follows the work of American pragmatism philosopher Dewey among others (Clancey, 1995), who claim that people are more inclined to learn by actively

participating in the learning experience. With situated learning, Lave and Wenger (1991) build on the idea that the community of practice is functioning as a framework for social learning (Lave and Wenger, 1991)**,** where learning takes place through social interaction and collaboration, suggesting that both new and former members are actively engaged contributors to the community.

2.3. Pragmatic view on the meaning of 'problem'

Pragmatism was a philosophical tradition that originated in the United States around 1870, whose core was the pragmatist maxim, a rule for clarifying the contents of hypotheses by tracing their "practical consequences" (Pragmatism | *Stanford Encyclopedia of Philosophy*, 2013). The American psychologist and philosopher John Dewey (1859-1952) was one of the most prominent exponents of pragmatism philosophy in the early 20th century, who had strong influence on educational reform and on the development of the so called "experiential learning theory". According to Dewey, learning as a term comprises objective external and subjective internal conditions, which he calls "situation" and "interaction" respectively. The external conditions are those surrounding us, like the academic authority or subject matter, and the internal conditions are the feelings of the participants, their inclinations, abilities and interests. In order to create better learning conditions, it is necessary not only to look at the more accessible external conditions, but as much as possible to look at them in conjunction with the internal subjective conditions. Dewey believes that a learning situation can never happen without a context. Everything we do happens in an interaction between the external and internal conditions, therefore an experience is always the result of an interaction between the individual and his environment at the time the event took place (Leahy, 2009).

Stemming from core ideas of Dewey, Aaen (2016) discusses thoroughly the pragmatic view on problems - what they are and how they are resolved. He states "*the moment we become conscious of a problem we also become conscious of the components that belong to the situation now perceived to be problematic"* (Aaen, 2016), where a situation is "*meaningful context within which objects and events are experienced"* (Dewey, 1938). Dewey opposed the notion that knowledge can be developed through abstract propositions as the formal logic prescribes. Instead, he claimed that knowledge is always created through "inquiry" into problematic situations. Dewey describes "inquiry" as a process that begins with the feeling that something is wrong. This feeling does not necessarily occur on the basis of an intellectual wit, but rather intuitively sensing that there is a problem. Only when the inquirer defines and formulates the problem, inquiry moves into an intellectual field by making use of the human abilities of

reasoning and thinking. In other words, inquirer can make use of his previous experiences with similar situations. Then, according to Dewey, the inquirer will try to solve the problem by testing a solution. In order for the problem to be resolved, however, the initial sense of uncertainty that led to the inquiry process to be initiated must have disappeared (Dewey, 1938). Recognizing problems is therefore a step in a process of transforming a problematic situation through inquiry into a determinate situation, which on the other hand, "*is a situation where our uncertainty and doubtfulness is resolved and replaced by a closed, finished, and unified situation*" (Aaen, 2016).

Taking into consideration the pragmatic view of "problem" and "problematic situation", it is obvious that a "problem" can imply different meanings depending on the context and the understanding of the experienced situation. The nature of crowdsourcing is about facilitating problem solving through an online platform. In that sense, developers who are part of the Topcoder community are solving design, development or data science problems posed by its customer organizations. Nevertheless, the crowdsourcing process itself can be challenging (Kondov, 2017), and put participants in a problematic situation. In the current study we are interested in recognizing the most pervasive problems posed by community members that relate to their participation on the crowdsourcing platform. As noted earlier, problem recognition is important and inevitable step in the process of transforming problematic situation through inquiry into a determinate situation. Therefore it can be argued that the goal of our inquiry is to achieve a determinate situation as an outcome of our observations. In line with these considerations, only through situation determination it would be possible to suggest a relevant solution or acknowledge whether such has been provided by the community management of Topcoder.

3. Method

This thesis utilizes an observational netnographic approach, described as an unobtrusive and less immersive online version of traditional "participant-observational" ethnography. Netnography has its roots in marketing, initially intended to adjust the traditional ethnographic research techniques and to apply them to consumer research conducted online. Since then, netnography has been widely used in various domains to study online communities and cultures. As the focus of this study are the problems experienced in a crowdsourcing online community of developers, we chose netnography as a method that relies primarily on online fieldwork (Kozinets, 2010). The following sections describe the method itself and how it has been used in more details.

3.1. The method of Netnography

Netnography was first introduced as a research method in the late 90s by marketing professor Robert Kozinets. Kozinets (2006) defines netnography as *"a qualitative, interpretive research methodology that adapts the traditional, in-person ethnographic research techniques of anthropology to the study of the online cultures and communities formed through computer-mediated communications".* The main purpose of the method is to enable the gathering of consumer and marketing insights by identifying and understanding relevant consumers' needs, beliefs, attitudes and motivations through their interaction in online communities on the Internet (Kozinets, 2002).

As netnography could be seen also as "ethnography conducted on the Internet", we should understand what exactly ethnography is. Hobbs (2006) defines ethnography as "*a cocktail of methodologies that share the assumption that personal engagement with the subject is the key to understanding a particular culture or social setting. Participant observation is the most common component of this cocktail, but interviews, conversational and discourse analysis, documentary analysis, film and photography, life histories all have their place in the ethnographer's repertoire*."

The primary benefits of the netnographic method are that it is less time consuming and less costly compared to traditional ethnography, as it is not dependent on the physical boundaries in the same way as traditional ethnography, where the researcher is physically present along with the object that is being studied. This assumes strong engagement by the researcher as he relies to big extent on participant-observational field work to gather data. In comparison, netnography uses existing empirical data available on the Internet, through online discussion forums, social

network websites, news boards, documents, etc. Although some degree of immersion and engagement is expected here as well, the web based character of the research allows for more passive involvement and rather observational analytical approach. This has an advantage that the netnographic participant-observational nature is less obtrusive than traditional ethnography and it is avoiding the bias that could be present when using interviews, surveys and focus group methods, as it is conducted using observations of community members in a context that is not fabricated by the researcher (Kozinets, 2002).

Kozinets (2010) points out that netnography can supplement a traditional ethnographic study with a varying degree of integration and coordination between the two. He makes a distinction by referring to the difference between a research on an online community and research on a community online. Often a social phenomenon might extend its presence from real world to an online setting, but usually, when the culture or community that is being studied, is present exclusively in an online domain, its nature leads to the choice of netnography as a standalone research method.

3.2. Research design of the study

Similar to ethnography, netnography is conducted through a specified set of methodological procedures and protocols that have been agreed upon by a community of scholars and like ethnography has inherent and necessary flexibility (Kozinets, 2010). Therefore, netnography usually follows in the six steps common for ethnography: research planning, entrée, data collection, interpretation, ensuring ethical standards, and research representation.

The current study will generally follow the guidelines by Kozniets (2010) with a few minor adaptations. Costello et al. (2017) acknowledge that it is a common practice among researchers to either adapt or omit particular netnographic steps to suit their study design. The first two steps “research planning” and “entrée” were preceded by a study on the inherent benefits and challenges of using crowdsourcing in software development (Kondov, 2017). This directed the current inquiry further toward choosing the community of Topcoder to be researched, as it is the most popular crowdsourcing software development platform at the moment.

In a qualitative study, Creswell (2009) advises toward formulating a broad research question that asks for exploration of the central phenomenon by examining the factors surrounding it and presenting the different perspectives of participants experiencing this phenomenon. This seems a good fit to netnography as it assumes a more open exploratory approach in line with the novel context of online communities and culture, but it can be argued that narrowing the research

question might be beneficial as the phenomenon is not considered brand new and there is existing research on the topic (Kondov, 2017).

The author of this paper has registered as a user on the Topcoder platform in May 2016 due to increasing interest in the crowdsourcing software development phenomenon. Nevertheless, he was not actively involved in the community until the commencement of the current project and therefore has taken primarily an observational stance. The observational approach implies that the interactive, conversational data from online communities should be treated as qualitative data to be "content analysed" (Kozinets, 2010). This approach was chosen, as active researcher participation in netnography has been questioned by some scholars (Langer and Beckman, 2005). Langer and Beckman (2005) advise that it might be beneficial for studies of online communities if the researcher does not expose himself in order to avoid endangering the unbiased outcome of the study. This relates also to the ethical guidelines to be followed, which according to the authors can be less rigorous when using publicly available data.

We have discussed so far the reasoning behind the first two steps of this study. The subsequent phases of data collection, data analysis and findings discussion will be presented further, after a brief review of Topcoder, the selected crowdsourcing software development platform. Figure 1 shows the current netnographic research process represented diagrammatically, although much cleaner than the way it occurs in reality.

Figure 1. Simplified flow of the netnographic research project, based on Kozinets (2010)

3.3. Crowdsourcing software development - the case of Topcoder

Topcoder is a United States based software development crowdsourcing platform, founded in 2001 by Jack Hughes. When Hughes felt that things were not going that well in his technology consultancy firm, he came up with the idea of using programming competitions as a way to keep his employees engaged in the working process. After selling his company, convinced that those competitions have greater potential, Hughes got interested in what the software development work space might look like in the future and he realized that they have touched upon many work related issues in his previous company.

Topcoder has been founded with Hughes's vision that the nature of work will change dramatically in the future and work that provides some degree of freedom and is done much more like play, will keep people interested, more productive, and happier with what they do. That vision is reflected in the aim of Topcoder to create and maintain a diverse virtual community of developers, who have the freedom and are encouraged to use their inherent talents and skills to solve problems in many different problem domains for many different companies. Collaboration by competition within the community is a main theme in Topcoder's management, as projects are cut down into small pieces allowing developers to work simultaneously on each part of the project. Developers are competing to provide the best solution and no one is concerned with who is working on a particular part, as long as the output of the competition is of high quality (*Collaboration by Competition | Business Innovation Factory*, 2017).

The innovative business model of Topcoder went through several transformations. At first, the platform served as a recruiting center where companies go to find programmers who are proven to be highly skilled, and where talented programmers show their skills to a worldwide audience. Large companies would sponsor Topcoder competition events creating opportunities for talented developers to come forward. Hughes then decided to use the community to redesign and recode Topocoder's own platform, which proved that it is possible to build high-quality complex software systems cost and time effectively using crowdsourcing (Lakhani, 2010).

Initially, Topcoder adopted a model aimed at creating solutions for clients by contracting with community members, running competitions, and providing consulting services. Not long after, the company created a catalog of reusable components from the software it was creating, which became an important part of Topcoder's value proposition to

its clients. Revenues grew because of this hybrid consulting model, but Hughes was still dissatisfied as platform management costs remained high. Therefore, by early 2009, Topcoder has moved away from this hybrid consulting model and focused more on executing software development tasks through competitions, where clients had to pay a monthly fee based on software requirements complexity and number of expected competitions required, instead of paying for the time of Topcoder's platform managers. The platform fee would also grant unlimited access to the increasing number of components in the Topcoder's catalog of reusable software components (Lakhani, 2010).

In September 2013, Topcoder was acquired by a company called Appirio, creating a community of more than 600,000 developers, engineers and designers, collaborating on software development tasks ranging from applications and websites to back-end enterprise systems. Appirio is a global information technology company that specializes in providing its clients with actionable strategies using the latest cloud technologies, relying on Topcoder's crowdsourcing community and thousands of pre-built solution accelerators (*Company Overview – Appirio* | Appirio, 2017). In October 2016, Appirio was acquired by Wipro, an IT leading company with a global reach in more than 175 cities. Wipro is now committed to the idea of using crowdsourcing to drive innovation within their customers, their partners, as well as internally, thus opening great new opportunities for Topcoder (*News Item: Wipro Acquires Appirio* | Topcoder, 2016).

Since its inception in 2001, Topcoder's community has grown exponentially. By spring 2009 it had around 200,000 members and currently the number is over 1 million, although it is believed that the actual number of active participants is a lot less. Registration process is very easy and everyone around the world with the needed skills who has a computer and access to the Internet can register through a simple form on the website and start competing on software development challenges right away.

Topcoder is facilitating competition in four primary areas - design, development, data science and competitive programming. These can be further divided into weekly organized crowdsourcing challenges within design, development and data science, and competitive programming algorithm matches known as Single Round Matches (SRMs) that are held twice a month. It is the first type that requests from community members to work on Topcoder's clients' projects through the specified challenges and rewarding them with monetary prizes. The second type offers competitions for solving demanding and complex algorithmic problems

where the aim is to earn peer recognition through receiving higher and more prestigious rating in the community.

On Figure 2. is shown the typical workflow for participation in a Development challenge as described on the Topcoder's website. Design and data science challenges follow the same structure to large extent with some minor variations. To determine winners and assess quality in client software development, Topcoder uses a community-based peer review system. Experienced Topcoder community members are paid to grade and comment on all contest submissions using detailed scorecards, ultimately picking the contest winners. The winning competitors for each contest are then awarded with monetary prizes and all participants are given updated ratings for their performance (Lakhani, 2010).

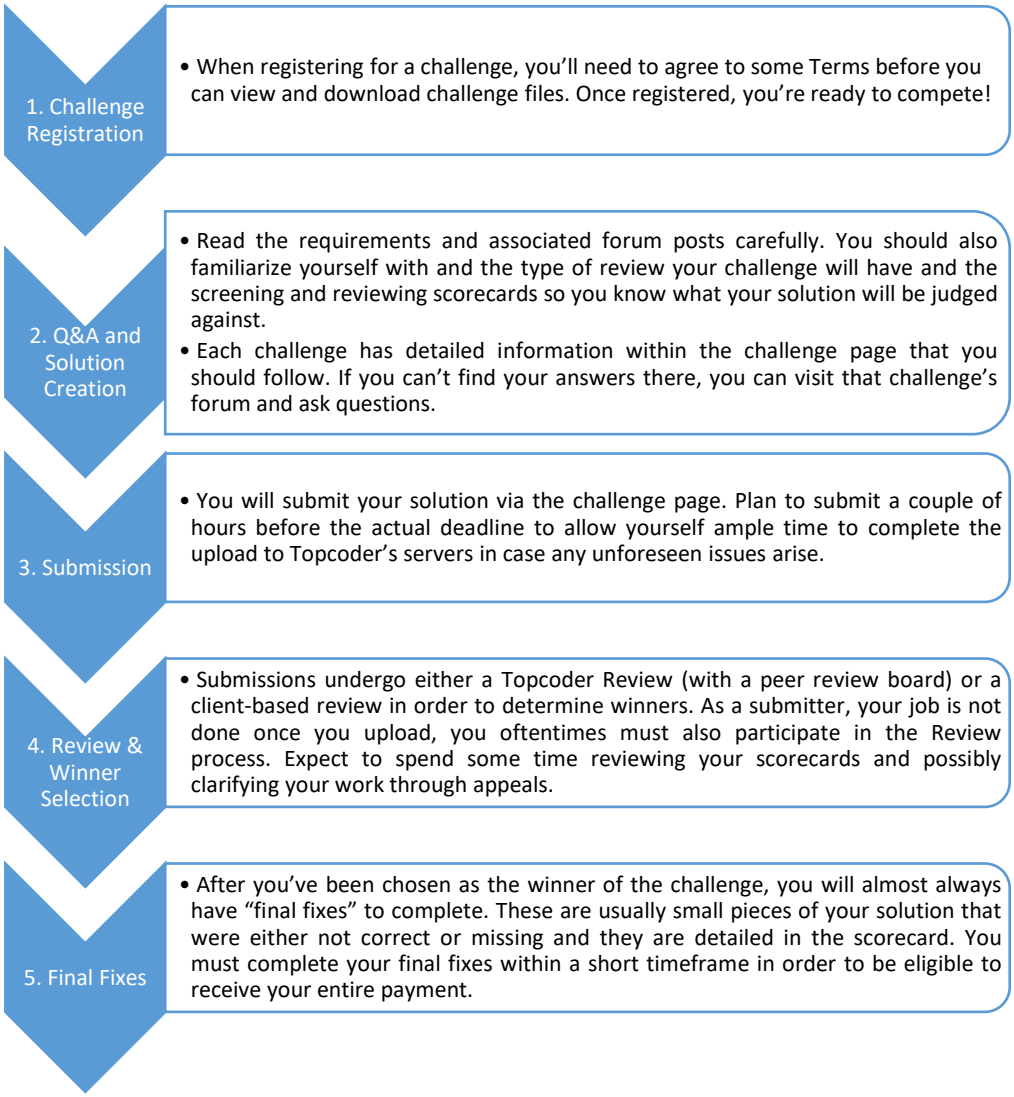

Figure 2. The process of participating in a development challenge on Topcoder (Source: Topcoder.com)

3.4. Data collection

In netnography, data collection and data analysis are closely tied together, varying on a spectrum of engagement in the online community from purely observation to extremely involved participation. Content analytical approaches take the observational stance of netnography, while the participative element is brought forward by expressing subjective personal reflection or auto-netnography. Ideally, the researcher has to constantly maintain tension between the reflexive and subjective mode of the engaged participant with the scientifically objective observer (Kozinets, 2010). In the process, the netnographic researcher can collect three different types of data: archival, elicited and filednote data.

This study will rely mostly on archival data, collected from the Topcoder's discussion forums (accessible at https://apps.topcoder.com/forums). We are interested in the main forums about the different disciplines of Design, Development and Algorithm, but also the Round Tables forum where the entire community can connect and discuss various topics regarding the platform. Topcoder uses also a community chat-like Slack group channels to enable easier communication between members. The Slack channels were included as a data source for the study, but with caution, due to some existing limitations. First of all, the Slack community channel is not that easily accessible to all community members, it is required to apply for admission to one of the community management team members and wait for approval. A few days after initial request, the researcher gained access, but the current subscription plan of the community allowed for only a limited number of messages to be shown, limiting access to archival data. Furthermore the channels were heavily moderated and it was observed that members were asked to post in the forums in case an issues requires attention by the platform community management, while Slack was intended primarily for quick communication with project co-pilots.

As noted earlier the author of this paper was registered as a user on the Topcoder platform in May 2016. The temporal span for the data collection procedure was decided to cover approximately one year before, including forum messages dating back to the beginning of 2015. This was decided because the researcher found out that the Topcoder website underwent a major redesign back then and some parts of the forums, like Design Discussions for example, have been created at that point of time. Also it was considered a reasonable time span in order to acquire a valid snapshot of past and current discussions about experienced problems and how are or were they being addressed by community management.

4. Data analysis

According to Kozinets (2015), "*the theoretical power of netnography flows from a search for anomalous evidence that must include deductive and inductive, as well as abductive reasoning*". This statement assumes looking deductively for particular keywords and key concepts, and classifying and coding inductively the miscellaneous pieces of data. The approach to analyzing qualitative data, will comprise the entire process of turning the different collected observation materials into a finished research product. This encompass using content analysis and interpretation to process and refine raw data to its essence, providing basis for discussion and creating new knowledge for the reading audience.

4.1. Data analysis approach

With the acquired pragmatic understanding of a problematic situation, the author will keep focus on identifying existent and emergent problems from the collected data through rigorous content analysis. A conventional content analysis approach will be adopted as described by Hsieh and Shannon (2005), which, according to the authors, is suitable when existing theory or research literature on a phenomenon is limited. This approach assumes allowing insights on categories and names of categories to emerge by immersing into the data, instead of using preconceived categories, and subsequently addressing relevant theories or other research findings in the discussion section of the study (Hsieh and Shannon, 2005).

Considering the problem solving character of a crowdsourcing platform and the understanding we gained to the meaning of "problem" from a pragmatic perspective, it is necessary to make some distinctions and draw boundary lines as to what would be considered a problem relevant to our study. As noted earlier, in a general sense there are different meanings associated with the term "problem". There are problems posed by client companies, which are tasks requested by a client looking for solutions, but there are problems related to the crowdsourcing process, experienced by community members. It is the latter type we are interested in. We want to see which problems are most pervasively raised and how are they being addressed by community management.

4.2. Findings

This section reports and exemplifies the major findings of the conducted research in order to illustrate what are the most pervasive problems raised in the community of Topcoder, and how is the community management trying to address them. Further discussion of the research findings is presented in the next section of the study.

4.2.1. Most pervasively raised problems in the Topcoder's community

The most pervasive problems that emerged from analyzing the messages in the discussion forums and Slack channels can be categorized in three main areas, namely: 1) technical problems with the Topcoder platform, 2) problems related to the usability with the Topcoder's website, and 3) problems related to the workflow process at Topcoder.

4.2.1.1.Technical problems with the Topcoder platfrom

Largest portion of the problems raised in the discussion forums relate to technical issues with the platform. These are about different types of bugs in the system that prevent community members to perform the tasks they want to, for example getting an error when trying to upload a solution to a challenge or not being able to log in to the platform.

A thread dating back to 2012 reports about a problem with logging in to the Topcoder Arena, even though logging in to the Topcoder website worked fine. The same problem has been raised by multiple members of the community spreading over the years until the end of 2016, when the bug probably got identified properly and fixed. User named "*LiChenKoh*" was very explicit when describing the problem:

"*I am not able to log in today to the Topcoder Arena Java applet. I was able to log in last night, and did not change any internet settings in between.*"

Most recently a number of users have been reporting in the Slack support channel about receiving a "*504 Gateway Time-out*" error when trying to reply in the forums. The user "*talesforce*" described the problematic situation:

"I've switched to multiple gateways, incognito window, cache clearing and I see same issue with all. Is anyone actually able to post in forums?"

Some members confirmed having the same issue, while others did not. For a few days the Topcoder's support has been trying to fix the issue probably caused by problems with its web servers, or as one of the community members suggested, it could be related to the authentication services API of the platform. Maybe because of the frequent bugs, recently "*tonyj*", a community management member, notified everyone in the Slack channel that they are getting close to replacing the submission management page with a new one.

There have been also discussions about a recent decline in Topcoder's popularity, and more specifically participant's interest in its competitive programming part. User "*niyaznigmatul*" has brought into the forums a discussion from another website about competitive programming (*www.codeforces.com*), asking the question "*Why is Topcoder dying*?". There it has been noted

that the current number of match participants seem to be at around 20% in comparison to what they used to be just a couple of years ago. The main issues in the Algorithm track according to "*niyaznigmatul*" are as follows:

- *"Bad web arena, it's too buggy (i.e the problems in standings are not sorted by points) and it doesn't support plugins"*
- *"Very connected to first issue, Java arena has some security difficulties running it: one cannot run it, if he can't turn off Java security checks and warnings"*
- *"Site is awful, you cannot find any information on it"*

Following up on the same discussion, some community members attribute this to ineffective management of Topcoder's platform or change in priorities for its future direction of development. Being one of the more active users, "*dimkadimon*" shares his view on the current situation and adds further to the previously noted issues:

- *"Algorithm problems got too hard, […] many people would score 0 and their only hope was to get some successful challenges. SRMs became less fun."*
- *"The TopCoder rating system is suboptimal and has deflation issues. It is possible that people left, because it became too hard to improve one's rating."*
- *"Some of the original TopCoders have retired and moved on to other things in their life"*
- *"A few other great programming sites have cropped up (like Codeforces, HackerRank and Codingame) and many members moved over."*
- *"Admins have stopped listening and only promise. In the past, admins cared for the community and they would frequently implement features that the community wanted. It is not the case now."*
- *Due to above issues, the forums have dried up. Little forum activity discourages users from staying with the site."*

4.2.1.2. Problems related to the usability of the Topcoder's website

It is reported that the website of Topcoder underwent major redesigning and revamping in the beginning of 2015. There are a number of threads discussing the new design of the website and the prevalent opinion is that it is not as good as the old version. In January 2016, approximately a year after the redesigned website launched, the user "*dimkadimon*" went even

further by creating and posting an online survey in the forum (results are accessible at: https://en.surveymonkey.com/results/SM-3K8ZBVZJ).

To summarize the insights gained from his survey, in total 52 members took part and the results confirmed the general opinion. Almost 85% answered that they prefer the older website design and find it difficult to navigate in the new one. Furthermore, large portion often find bugs on the website and think that the time for the bugs to be fixed takes longer than a month. The majority of members found the new website to be too slow, almost half of them hated the visual aspects of the design, and more than half of them answered that their level of website usage decreased over the last few years. It is arguable how valid and reliable are the results of this survey, but nonetheless they have been a big red flag for the Topcoder community management.

A Topcoder staff member "*mess*" has entered the discussion explaining that they try to improve, even though it is difficult to balance between new and old members, who have different interests in solving development, design or algorithm challenges:

"For instance in algorithms, we can look at upgrading our content around editorials and tutorials as well as other things. I'd love to hear more feedback around more items the community thinks we need or can improve on that the community can help produce!"

Contrary to the survey results, another community management member has noted that in every other SRM challenge they ask the participants a simple question: whether they are willing to recommend Topcoder. Around 77% answered positively during the course of 2015. Nevertheless, she tried to reassure members that the results from this survey matter to the community management and they are trying to listen and evaluate feedback from all sources available, as they are trying to increase satisfaction across the community.

4.2.1.3. Problems related to the workflow process at Topcoder

A community member with nickname "*billsedison*" raised a problem in the Slack channel concerning the fairness of the reviewing process in Topcoder's challenges, creating an interesting discussion among fellow community members. He posted the following question:

"Suppose a member is eligible as both competitor and reviewer, he joined a series of challenges as a competitor. But in some other challenges of this series, he played the role of a reviewer. It seems to be a common case, but do you think it is acceptable or fair?"

Although members familiar with particular series of challenges are obviously more suitable to be reviewers, they might be biased. According to "*billsedison*", if a member has failed to win the 1st place or prize in the previous series of challenges and now has joined as a competitor, there would be no guarantee that he will judge objectively the submission of other participants, who were his competitors in the previous challenges. Furthermore, this member would also have an advantage if he decides to participate in the following challenges of these series.

Another area for common problems raised by community members was concerning payments of won prizes from Topcoder. Currently, the platfrom supports three payment methods: PayPal, Payoneer and Western Union. A message from a member with nickname "*tinasatija*", regarding an error in the system, was originally posted in August 2016 and same problem was confirmed by another member in August 2017. This thread has not received any response from Topcoder's staff members, but seems that this was not a common practice. Many similar problems have been reported previously with the other payment methods Payoneer and Western Union. Those were related to delay in transfers and rejected payments. It does not become clear whether the problems come from Topcoder's or the financial institution's side, but obviously members get frustrated. For example, user "*rainforest*" reports a 4 months delay of her payment from Paynoeer, who does not even reply to her requests:

"*I am really frustrated with Payoneer and their service. Hope you review your payment provider recommendations so that other members also do not suffer similar experience. Please help."*

She has received a response from a community member who ensured her that he would deal with the case through the support ticket system, and this seems be the common practice most of the times when similar issues are raised in the forums.

Many members have experienced also problems related to taxation when receiving payments from Topcoder. There are different rules depending on the country you live in and the way to report income received from online activities varies. Therefore it is up to the members themselves to correctly work out taxation process. There are a number of threads asking for help to properly handle these issues, but they are usually left unanswered probably as Topcoder is not directly responsible for solving them.

4.2.2. How is the community management of Topcoder addressing raised problems?

In 2014 Topcoder has established a so called Community Advisory Board (CAB) in order to better follow what problems are raised by the community and to work more effectively for solving them. There are CAB meetings held every month to discuss issues and potential improvements in all Topcoder tracks. CAB is represented by a team of six community members, two from each track (Development, Design, and Data science/Algorithm), who are particularly interested in the future development of Topcoder. They serve as a connection between the community and the Topcoder staff, bringing community ideas and concerns to their attention, and helping in the implementation of new features and platform enhancements (*Topcoder Community Advisory Board* | Topcoder, 2017).

Members of the CAB are active and can compete on challenges or act as co-pilots and reviewers. They are elected for a 12 months long period in the beginning of the year and have the following responsibilities (*Topcoder Community Advisory Board* | Topcoder, 2017):

- Attend monthly meetings with Topcoder staff
- Review new ideas and features and provide feedback to Topcoder staff
- Regularly communicate with the members and prioritize all wants/needs of the community
- Recap meetings in a blog post and in the CAB forum
- Maintain active voice in the CAB forum

In January 2017 CAB members came up with a new idea about using GitHub to tag issues and collaborate to solve them. All the CAB members get an invitation to join the Topcoder CAB team on GitHub, where they can create issues and tag them with the appropriate label according to respective track. CAB members can start communicating with the Topcoder members, gather information for the improvement of the community and write down issues, thus the team can discuss them right away. The process resemble to big extent a sprint cycle burndown chart where there are planned tasks, tasks that are being worked on, and tasks that are already finished.

5. Discussion of the research findings

Going through the Topcoder forums it became clear that they are not very active overall, especially in some specific sections. In contrast, when it comes to problems with the website or platform, there are some community members that take active participation in the discussions. Generally, most of the times threads with a substantial number of replies contained more valuable information for the study. Some threads that were created a few years ago, would be put forward again by a recent poster, having similar issue. This sometimes made it confusing whether it is really a currently important problem, but other times it helped to track how pervasive was the problem over time and what actions were taken from the community management members.

Most of the problems raised were related to the technical and usability aspects of the Topcoder platform. There is prevalent opinion among community members that the current design of the website and the platform is not good enough. Older, as well as more recent posts, report for some occasionally experienced bugs in the system. The poor design as well as the buggy platform system are pointed out by a number of community members as the main reason for the decline in participation activity, especially in the competitive programming area. It was interesting to see actually, that there were not so many complaints about problems with the Development and Design tracks of Topcoder. This should make us think, in line with some community members' comments, that maybe Topcoder is increasingly changing its focus toward the more profitable part of its business. Majority of the members who expressed their opinion in the forums find the current situation to be quite regrettable, as Topcoder has been created with the intention of being a competitive programming platform. All these problems seem to be influencing factors to the decreased motivation, especially among some long standing members of the community.

It is admirable that the Topcoder community management seem to be quite open and responsive to problems raised in the community. But it made an impression, that they are not very consistent at times with their responses and sometimes there is a big delay or they are just lacking. This is quite surprising actually that some issues require so much time to get resolved, especially given that we are talking about a company dealing with software development. A possible reason for this might be existing difficulties with migrating the old website, database, servers, etc., or an ongoing organizational restructuring. After all Topcoder went through two big acquisitions in less than five years.

Before commencing the current study, it was speculated whether and to what extent the problems experienced by workers would be interrelated with those from an organizational perspective, especially considering that the platform as an intermediary plays a major role in the crowdsourcing process. Findings from the current study show that developers raise different problems in the forums, compared to what one would expect to face as a problem in a crowdsourcing organization. The reason for this might be that the discussion forums are mostly about developing the platform and the community, while the customer organizations are more concerned with problems from their own organizational managerial perspective. Nevertheless, if we look at the findings from a higher level of analysis, they all relate in some degree to the motivation of community members to participate in the crowdsourcing platform. Keeping high motivation is indeed an important challenge for every organization, including those considering to use crowdsourcing software development.

Topcoder is a good example of a crowdsourcing platform facilitating open innovation. More specifically open innovation in a software development domain. It provides flexibility to its clients who have the opportunity to tap into a community possessing enormous intellectual potential outside of their organizational boundaries. They can rely on a pragmatic approach to problem solving as Topcoder employs proven methods for acquiring the best solution possible to challenging software development problems. It is the community that makes Topcoder's value proposition so attractive and the practice of software development that binds the constituent parts together. But is Topcoder's community really a community of practice? Probably the answer to that question would depend on the perspective we take as the concept has evolved quite a lot in the last two decades. Taking the later perspective of Wenger et al. (2002) it certainly is, as the members of Topcoder's community share the same concerns, problems, and passion about software development, design, and data science. They deepen their knowledge and expertise in these areas through ongoing participation and interaction.

Putting an end to the discussion section we should acknowledge that there are some limitations to the current study. It might be argued that taking a more active and participative stance toward data collection would have benefited the resulting findings, nevertheless the author followed established methodological procedures and the results can be considered to be valid and reliable. The short time frame for the study to be written and the fact that it was conducted by only one researcher has to be taken into consideration as well.

6. Conclusion

Crowdsourcing software development is an innovative and emerging approach to software development. The current paper builds on previous study conducted by the author which synthesized available literature on inherent benefits and challenges of using crowdsourcing in software development. The perspective was shifted from the platform's clients to its community of developers. The reasoning behind this is that in any business organization, its workforce is considered to be a highly valuable asset and in order to be successful, it has to listen to their problems and needs. The purpose of this thesis was to gain deeper understanding of the Topcoder's community of developers, more specifically it was aimed at identifying the most pervasively raised problems related to participating in crowdsourced software development, and uncovering how Topcoder's community management addresses these problems.

The netnographic method used proved to be beneficial for directing the qualitative research of Topcoder's online community. With using conventional content analysis of the collected data, three main categories of problems emerged as being most pervasively raised by community members, namely:

1) technical problems with the Topcoder platform

2) problems related to the usability of the Topcoder's website

3) problems related to the workflow process at Topcoder

Technical problems with the Topcoder platform take largest part among raised problems in the community. These are related to different types of bugs experienced by members when using the platform. For some members, there are serious issues existent that would prevent them from participating in challenges or in competitions. To some extent these can be related to the second category of usability problems with the Topcoder's website, which are being raised in the community. These relate to poor design and usability of the website, which is putting off members who otherwise wish to participate more actively. The third category of problems is related to the workflow process at Topcoder. Most prevalent are problems experienced by members wish to withdraw the payments they earned, or problems related to some specific part of the workflow process, such as ensuring transparency when reviewing challenge submission. There is a dedicated team of expert community members called CAB that is trying to address the problems raised in the community, but the effectiveness of their actions is questionable.

As final remarks, findings from this study can have implications for Topcoder or similar platforms who wish to improve their design and focus on specific problematic areas that cause strong demotivation, discontent and dissatisfaction among community members.

References:

Aaen, I. (2016). *Essence: Pragmatic Software Innovation*. Unpublished book draft. Department of Computer Science, Aalborg University.

Appirio.com, (2017). *Company Overview – Appirio* | Appirio. Available at:

Bergvall-Kåreborn, B., & Howcroft, D. (2014). Persistent problems and practices in information systems development: a study of mobile applications development and distribution. *Information Systems Journal, 24(5),* 425-444.

Brown, J.S. and Duguid, P. (1991) Organizational learning and communities of practice: toward a unified view of working, learning and innovation. *Organization Science,* 2(1), 40–57.

Chanal, V. and Caron-Fasan, M. L. (2008). How to invent a new business model based on crowdsourcing: the Crowdspirit® case. *In: EURAM Lubjana, Slovenia.*

Chesbrough, H. (2003). *Open Innovation: The new Imperative for creating and profiting from technology*. Harvard Business School Press, Boston, MA.

Chesbrough, H. (2006). *Open Innovation: Researching a New Paradigm*. Oxford University Press.

Cohen, W. M. & Levinthal, D. A**.** (1990). Absorptive Capacity: A New Perspective on Learning and Innovation. *Administrative Science Quarterly*, vol. 35, issue 1, pp. 128-152.

Business Innovation Factory, (2017). *Collaboration by Competition | Business Innovation Factory*. Available at: http://www.businessinnovationfactory.com/summit/story/collaboration-competition

Clancey,W.J. (1995). A tutorial on situated learning. *Proceedings of the International Conference on Computers and Education* (Taiwan) Self, J. (Ed.) Charlottesville, VA: AACE.

Creswell, J. W. (2009). *Research design: Qualitative, quantitative, and mixed methods approaches*. Thousand Oaks, CA: Sage Publications.

Costello, L., McDermott, M-L., Wallace, R. (2017). Netnography: Range of practices, misperceptions, and missed opportunities. *International Journal of Qualitative Methods*, 16 (1), 1–12.

Cox, A. (2005). What are communities of practice? A comparative review of four seminal works. *Journal of Information Science*. 31(6) 527–540.

Dahlander, L. and Gann, D. M. (2010). How open is innovation? *Research Policy,* vol. 39, issue 6, pp. 600-709.

Dewey, J. (1986). The Reflex Arc Concept in Psychology. *Psychological Review 3*: 357-370.

Dewey, J. (1938). *Logic: The Theory of Inquiry*. Henry Holt and Company, New York.

Duguid, P. (2008) Prologue: Community of Practive Then and Now. In: Amin, A., Roberts, J. (eds) *Community, Economic Creativity, and Organization*. Oxford: Oxford University Press.

Hallerstede, S. H. (2013). *Managing the lifecycle of Open Innovation Platforms*. Springer.

Hasteer, N., Nazir, N., Bansal, A. and Murthy, B. K. (2016). Crowdsourcing Software Development: Many Benefits Many Concerns. *Proceedings of the 1st International Conference on Information Security & Privacy.*

Hobbs, D. (2006). 'Ethnography'. In V. Jupp (ed.). *The SAGE Dictionary of Social Research Methods*.

Howe, J. (2006). *The Rise of Crowdsourcing*, Wired Magazine. Available at: https://www.wired.com/2006/06/crowds/

Hsieh, H.F. and Shannon, S. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research* 15, 1277–1288.

Kondov, S. (2017). Benefits and challenges of using crowdsourcing in software development. *Semester project, Aalborg University*.

Kozinets, Robert V. (2002). The Field Behind the Screen: Using Netnography for Marketing Research in Online Communities. *Journal of Marketing Research*, 39(1), 61—72.

Kozinets, Robert V. (2006). 'Netnography'. In V. Jupp (ed.). *The SAGE Dictionary of Social Research Methods*.

Kozinets, R. V. (2010). *Netnography: Doing Ethnographic Research Online*. Thousand Oaks, CA: SAGE.

Kozinets, R. V. (2015). *Netnography: Redefined*. SAGE Publications, Ltd.

Lakhani, K. R., D. Garvin, E. Lonstein. (2010). TopCoder (A): Developing software through crowdsourcing. *HBS Case 610-032, Harvard Business School, Boston.*

Langer, R. and Beckman, S.C. (2005). Sensitive research topics: Netnography revisited. *In: Qualitative Market Research*, Vol. 8, No. 2, S. 189-203.

Lave, J., Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. New York: Cambridge University Press.

Lazar, J., Feng, J. and Hochheiser, H. (2014). *Research methods in human-computer interaction*. John Wiley & Sons, Inc.

Leahy, R. (2009). *Authentic educating*. University Press of America.

Morse, J.M. (1999). Qualitative generalizability. *Qualitative Health Research 9(1), 5–6.*

Nambisan, S. and Sawhney, M. (2007). A Buyer's Guide to the Innovation Bazaar. *Harvard Business Review*, (85:6), pp. 109-118.

Osterwalder, A. and Pigneur, Y. (2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. John Wiley & Sons, Hoboken, New Jersey.

Phillips, J., (2011). Open Innovation Typology. In Sloane, S. (ed.) *A Guide to Open Innovation and Crowdsourcing: Advice from Leading Experts*. Kogan Page Limited, Great Britain and United States

Rigby, D. and Zook, C. (2002). Open-Market Innovation. *Harvard Business Review*, vol. 80, issue 10, pp. 80-89.

Rothwell, R. (1991). External networking and innovation in small and medium-sized manufacturing firms in Europe. *Technovation*, vol. 11, issue 2, pp. 93-112.

Schenk, E. and Guittard, C. (2009). Crowdsourcing: what can be outsourced to the crowd, and why? *In: HAL: sciences de l'Homme et de la Socie´te, version 1–8.*

Stanford Encyclopedia of Philosophy, (2013). *Pragmatism | Stanford Encyclopedia of Philosophy*. Available at: https://plato.stanford.edu/entries/pragmatism/

Stol, K.J. and Fitzgerald, B. (2014). Researching crowdsourcing software development: perspectives and concerns. *ICSE '14 Workshop on Crowdsourcing in Software Engineering*.

Stol, K. J. and Fitzgerald, B. (2014). Two's Company, Three's a Crowd: A Case Study of Crowdsourcing Software Development. *Proceedings in the 36th International Conference on Software Engineering*

Surowiecki, J. (2005). *The wisdom of crowds*. New York: Anchor Books.

Topcoder.com, (2017). *Topcoder Community Advisory Board* | Topcoder. Available at:

https://www.topcoder.com/community/member-programs/topcoder-community-advisory-board/

Topcoder.com, (2016). *News Item: Wipro Acquires Appirio* | Topcoder. Available at: https://www.topcoder.com/blog/news-item-wipro-acquires-appirio/

von Wartburg, I., Rost, K. and Teichert, T. (2006). The creation of social and intellectual capital in virtual communities of practice: shaping social structure in virtual communities of practice. *International Journal of Learning and Change*, Vol. 1, No. 3, pp.299–316.

Wenger, E. (1998). *Communities of Practice: Learning, Meaning & Identity*. New York: Cambridge University Press.

Wenger, E., McDermott, R., Snyder, W.M. (2002). *Cultivating Communities of Practice*. Boston: Harvard Business School Press.