CATÓLICA LISBON
BUSINESS & ECONOMICS

The Impact of Contest Attributes on Solvers' Participation and Performance in Crowdsourcing Contests

Student: Fabian Kutschera

Supervisor: Cláudia Costa

Dissertation submitted in partial fulfillment of requirements for the degree International MSc in Busines Administration | Major: Entrepreneurship & Innovation at the Universidade Católica Portuguesa, 06.06.2016

ABSTRACT

Crowdsourcing contests as a means to leverage the knowledge of external individuals to solve firms' innovation problems are growing in popularity. Little research has been devoted to the question how such contests should be ideally designed in order to achieve a high level of participation and high quality solutions. By analysing a data set of 913 software engineering contests from the platform TopCoder, this study investigates the implications of three particular contest elements: the number of participants, the monetary award and the problem description. The results confirm the theoretic model, stating that a large number of participants is preferable for a seeker (firm) and outweighs the arising disadvantage of lowered efforts of solvers due to a greater competition. This theoretic discussion was enriched by the finding of a curvilinear relationship between the number of solvers and the contest's performance, wherefore adding contestants is only beneficial until a certain level. Furthermore, results show that a high monetary award captures more participants but leads to a lower contest performance, which is contrary to prior research. Regarding the problem description, it was shown that a short problem description increases participation and improves the contest performance whereas a problem description, which includes URL links to other homepages, reduces participation but improves a contest's best solution. By understanding the outcomes of this study, firms better understand how to design crowdsourcing contests.

Keywords: Crowdsourcing, open innovation, innovation, contest elements

RESUMO

Os concursos de crowdsourcing estão em crescente popularidade uma vez que representam uma forma de gerir conhecimento exterior às empresas para a resolução dos seus problemas inovativos. No entanto, pouca investigação tem sido dedicada a melhorar o design destes concursos para obter um elevado nível de participação e soluções de maior qualidade. Ao analisar um data set de 913 concursos de engenharia de software através da plataforma TopCoder, este estudo investiga as implicações de três elementos particulares do concurso – o número de participantes, o prémio monetário e a descrição do problema. Os resultados obtidos foram de encontro à teoria - um maior número de participantes é preferível. Esta relação mantém-se apesar do aumento do número de participantes diminuir os seus esforços para resolver o problema em questão (a competição é muito superior). Adicionalmente, os resultados obtidos atestam a relação curvilínea entre o número de participantes e o desempenho do concurso – o aumento do número de participantes só é benéfico até certo ponto. Ao contrário das conclusões obtidas em estudos anteriores, os resultados mostram também que um elevado prémio monetário atrai mais participantes mas dá origem a um pior desempenho. Quanto à descrição do problema, as conclusões obtidas indicam que uma descrição sucinta gera um maior número de participantes e melhor desempenho enquanto descrições com links para outras páginas dimuinem a participação melhorando, no entanto, a qualidade da solução obtida. Com este estudo pretende-se auxilar as empresas no processo de design de um concurso de crowdsourcing.

ACKNOLEWDGEMENTS

First of all, I would like to thank my supervisor, Cláudia Costa, from Nova School of Business and Economics for her constant support and encouragement during the process of defining the research topic and the guidance and positivity she abetted me with throughout the writing process.

Furthermore, I want to thank my parents for their support and also for funding my studies. Lastly, I thank my brother for reviewing this thesis and my friends in Lisbon and anywhere else who always support me.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In 1714, after a group of several famous scientists had failed to find a way to determine the location of ships in the sea, the British parliament announced a public tournament in order to find a solution in this way. The 'Longitude Prize' offered £15,000 to the person who would come up with a solution for this challenge. The winner of more than hundred participants was John Harrison, an English carpenter and clockmaker who had developed a highly accurate chronometer that made it possible to pinpoint a ship on the map (Boudreau & Lakhani, 2013). Besides being a milestone from an engineering point of view, the Longitude Prize is also one of the first documented crowdsourcing contests.

Crowdsourcing, a term coined by Jeff Howe (2006), broadcasts a problem to a large group of people in order to solve a firm's problem. By leveraging knowledge from external sources, the principle of crowdsourcing builds up on the concept of open innovation. Firms which innovate with the model of open innovation incorporate - in contrast to the model of closed innovation - external sources of knowledge, such as universities, suppliers or individuals (Chesbrough, 2003; West & Bogers, 2014). Several corporations from different industries use this model to innovate. For instance, the consumer product company Procter & Gamble defined the goal to find 50% of its innovations outside of the company's boundaries (Chesbrough, 2011).

As abovementioned, crowdsourcing contests, as one type of crowdsourcing, have their origin in the 18th century. However, the development got accelerated by the rise of the internet with its powerful and relatively inexpensive tools which facilitated firms and intermediary platforms to launch and manage crowdsourcing contests (Lampel et al., 2012). For example, the video platform Netflix launched one of the earliest online crowdsourcing contests in 2006. Participants were asked to improve the firm's algorithm for movie recommendations. The contest was a great success and the winning solution, that won the award of one million dollars, was able to improve the algorithm's performance by more than 10% (Boudreau, Lacetera, & Lakhani, 2011; Howe, 2006). Besides such ad-hoc contests, firms started to use crowdsourcing contests as an ongoing business model to source innovations (Boudreau & Hagiu, 2009).

Firms may either launch their own crowdsourcing contest platforms or, more commonly, publish their problems on existent intermediary platforms, such as InnoCentive[1]. Although crowdsourcing platforms evolved quickly and gained popularity over the last years, not much research on this type of crowdsourcing has been conducted (Lampel, Jha, & Bhalla, 2012; Leimeister, Huber, Bretschneider, & Krcmar, 2009).

Adamczyk and colleagues (2012) divided prior research of crowdsourcing contests into five perspectives: economic, management, education, innovation and sustainability perspective. Especially the ideal contest design, as part of the economic perspective, has not been fully explored yet and requires further investigation in order to answer the question on how contest attributes influence a contest's outcome (Adamczyk et al., 2012). Hence, this study aims to examine this gap in research and focuses on three contest attributes in particular: the number of participants, the price award and the problem description.

One major question in the economic literature discusses whether a contest shall be open to anyone or limited in the number of solvers. However, this has not been consensually answered. Several researchers argue for a limited number of solvers, as a result of the so-called "competition effect". This effect states that solvers in a contest lower their efforts when competing with a higher number of other participants. Thus, the average quality of submissions decreases. (Che & Gale, 2003; Garcia & Tor, 2009). Fullerton and McAfee (1999) even conclude that the ideal number of solvers is, in fact, just two.

On the other hand, a contrary theory, the "parallel path effect", argues that an unlimited participation is preferable for a firm that launches a crowdsourcing contest, since the resulting variance of solutions, in regards of quality, improves the quality of the best solution submitted (Dahan & Mendelson, 2001; Nelson, 1961; Terwiesch & Ulrich, 2009). The underlying assumption for this theory is that a seeker in an innovation contest is, in contrast to other contest types, only interested in the maximum solution (Girotra, Terwiesch, & Ulrich, 2010).

Terwiesch and Xu (2008) merged these two conflicting views and showed that the advantages of open access to contests can exceed the disadvantages of lowered efforts of solvers (Terwiesch & Xu, 2008). However, empirically, this theoretical model has not been fully examined yet. Furthermore, any potential curvilinear relationship between the number of participants and the contest outcome has not been researched yet.

[1] https://www.innocentive.com/

A second point of interest is the ideal reward mechanism. Winners of crowdsourcing contests gain a price award in form of money. Thus, participants are financially motivated which stands in contrast to other open innovation initiatives, such as user innovation, where users participate only for their status in the community or other intrinsic motivations (Afuah & Tucci, 2012; Leimeister et al., 2009).
Therefore, the monetary award, as a characteristic feature of a crowdsourcing contest, is a major discussion point in the literature (Cason, Masters, & Sheremeta, 2010; Shao, Shi, Xu, & Liu, 2012; Snir & Hitt, 2003). However, most of the literature analyses the impact of money as a motivation from the solver's point of view. This study, by contrast, aims to research the impact of the monetary award from a firm's point of view and explores how it affects participation and performance in a contest.
Finally, this study looks at the problem description. Little research has been devoted to this design element although it is the firm's initial and main point of contact with solvers. The problem statement explains the problem and the requirements that solutions must meet (Piller & West, 2014). Writing a problem statement is not an easy task. On one side it has to include all specific information so that the scope of interpretation is as limited as possible and information asymmetry between the seeking firm and the solvers is minimalized (Lüttgens, Pollok, Antons, & Piller, 2014). However, the initial example of the Longitude Price showed, that winners of crowdsourcing contests often have a field of expertise distant to the problem (Jeppesen & Lakhani, 2010). These solvers, who have a different knowledge background, also need to be able to convey the problem requirements wherefore a problem description has to be written in a simple and clear manner so that non-experts can also understand the problem (Afuah & Tucci, 2012; Spradlin, 2012).

Prior research has only considered the length of text of a problem description for empirical experiments. Yang and colleagues (2009) found that shorter descriptions attract more solvers to contests. However, this research paper considered a wide range of different crowdsourcing contest types and therefore lacks some concrete, practical relevance. Besides considering the text length, this study aims to include two other factors of a task presentation which are the number of URL links in the description and the readability of the problem text. These factors have been analysed in the field of crowdfunding campaigns (Greenberg, Pardo, Hariharan, & Gerber, 2013; Xu et al., 2014), wherefore it is obvious to also transfer this analysis to the neighboured field of crowdsourcing contests.

Thus, the conceptual framework of this research aims to answer how a crowdsourcing contest should be ideally designed, focusing on the design elements of the monetary award, the problem presentation and the number of solvers. (see Figure 1).

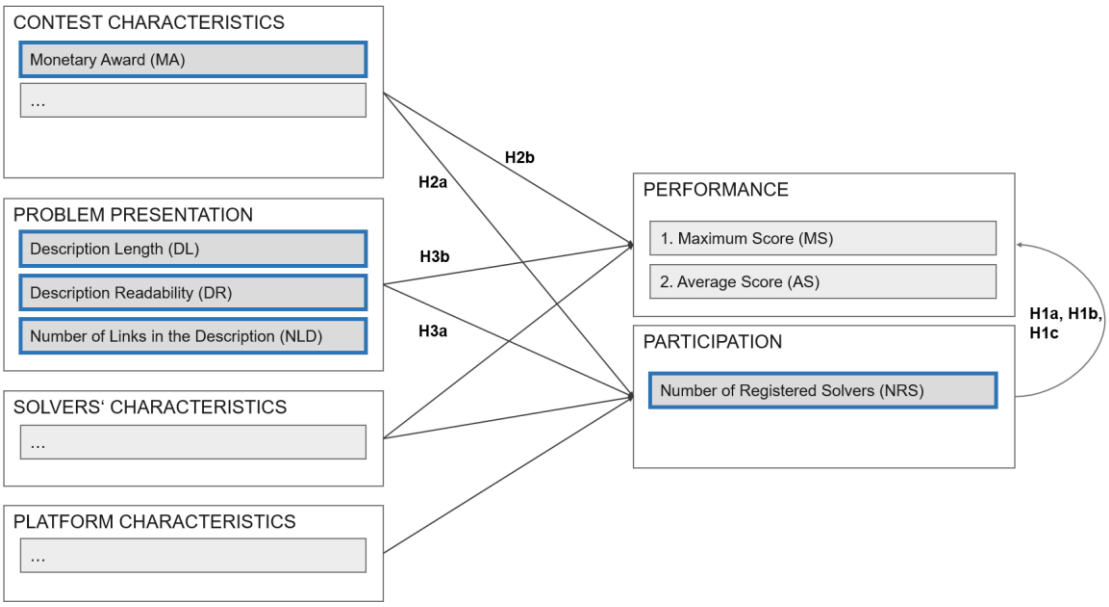


Figure 1: Conceptual Framework (Source: own work)

The research is based on a data sample of 913 crowdsourcing contests from the platform TopCoder[2]. The paper proceeds as follows: Section 2 reviews the prior literature and formulates hypotheses. Section 3 explains the research methodology. The empirical findings are presented in section 4, followed by a discussion in section 5.

[2] https://www.topcoder.com/

2 LITERATURE REVIEW

2.1 Open Innovation

For most of the last century, innovations were sourced exclusively in firms' R&D departments. This procedure guaranteed a certain level of quality, performance and availability of a technology (Chesbrough, 2003). Companies invested their profits into their internal R&D departments in order to find and develop the most innovative ideas and bring them to the market before competitors do (Chesbrough, 2011). Several firms like Xerox or IBM even commenced a vertical integration in the 1960s, meaning that all product components were produced by themselves due to a lack of suppliers for such components at that time (Chesbrough, 2003). This innovation approach, defined as the model of closed innovation, was used by firms in the twentieth century to innovate successfully (see Figure 2).

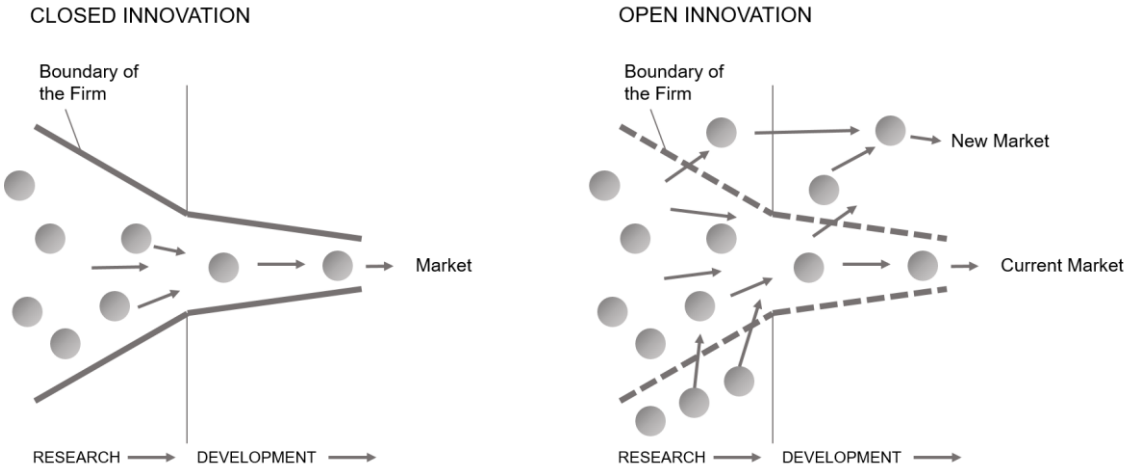


Figure 2: Closed versus Open Innovation Process (Adapted from Chesbrough, 2011)

However, the knowledge environment changed until the early twenty-first century (Chesbrough, 2003). Therefore, the closed innovation approach evolved over time into a more flexible and open approach of open innovation (see Figure 2) (West, Salter, Vanhaverbeke, & Chesbrough, 2014). The open innovation model, established by Chesbrough (2003, 2011), states that firms' ideas can be commercialized, not only by the firm itself, but also outside of the company in order to create additional value. Furthermore, the model argues that ideas from external sources can be brought into the company. Thus, knowledge flows between the firm and other parties are crucial in the concept of open innovation and the boundaries between the firm and its environment regarding the innovation process were starting to become weaker (Chesbrough, 2011; Mention, 2011).

Open Innovation is particularly promising for companies operating in industries that offer high levels of technological opportunities. This is manifested through higher searching costs for critical knowledge in such industries (Laursen & Salter, 2006).

Throughout the model of open innovation, firms which allocate limited efforts into internal R&D activities become capable of innovating by sourcing knowledge from external parties, providing that they are able to absorb this knowledge (W. M. Cohen, 1989). Such insourcing of external ideas can be beneficial for the company and is more and more used as a way of sourcing innovation (Mention, 2011).

Firms can integrate knowledge into the company from several external sources, such as customers, suppliers, universities or individuals (West & Bogers, 2014). One method of capturing knowledge of external individuals, is the method of crowdsourcing (Afuah & Tucci, 2012; Jeppesen & Lakhani, 2010).

2.2 Crowdsourcing

2.2.1 Overview

Crowdsourcing as a term was coined by Jeff Howe and is a neologism of the words "crowd" and "outsourcing" (Howe, 2006). Howe defines crowdsourcing as "the act of taking a task traditionally performed by a designated agent (such as an employee or a contractor) and outsourcing it by making an open call to an undefined but large group of people." (Howe, 2008, p. 2). Hence, crowdsourcing approaches broadcast a certain problem or task to a wide, heterogeneous network of external actors (Afuah & Tucci, 2012; Piller & West, 2014).

Although the usage of crowdsourcing was boosted with the rise of the internet, which enabled firms to reach out to a vastly bigger crowd, the crowdsourcing principle had already been used much earlier. As presented in the introduction, the Longitude Prize of 1714 was one of the first documented crowdsourcing contests. However, several other contests were held early in the history, for example the contest which Napoleon III announced in 1869, that offered a reward to anyone who discovers a process to produce a substitute for butter (Khan, 2005). These examples show that crowdsourcing was used already early in the history in order to leverage external knowledge for innovation.

Generally, firms are very interested in the model of crowdsourcing, since it allows to access external knowledge which the firm can use to solve problems it is facing. Many

companies quickly implemented crowdsourcing platforms although little results were known about their effectiveness. Moreover, since crowdsourcing communities do not exist for an extended period of time, there are not many insights into their successes (Bayus, 2013).

Nevertheless, existing research showed that external knowledge gained via crowdsourcing activities can be superior, compared with the knowledge sourced internally in a firm. Jeppesen and Lakhani (2010), who analysed problems posted on the contest platform InnoCentive, showed that firms were able to attain solutions for problems they previously had not been able to solve themselves. Moreover, through crowdsourcing a firm gains access to a large set of individuals with a high level of expertise. These solvers can be eventually exploited for recruitment purposes. When NASA launched a crowdsourcing contest, winners did not only win an award but were also invited to the firm and in some cases got hired (Lampel et al., 2012).

Research also revealed that crowdsourcing can be more cost-effective than internal innovation sourcing efforts (Afuah & Tucci, 2012). Malone, Laubacher and Johns (2011) show, for instance, that sourcing a solution on the crowdsourcing platform TopCoder only costs 25% compared to internally solving the problem. Furthermore, crowdsourcing activities on intermediary platforms, such as TopCoder, reduce financial risks since firms only pay for successful contests (Terwiesch & Xu, 2008). Thus, cost savings are another important reason why a seeker decides to release problems on crowdsourcing contest platforms.

Prior research also examined the users' motivations to participate in crowdsourcing activities in the first place. Brabham (2010) researched users of istockphoto[3] - a platform where users upload professional photos and get a profit margin for each download - and found that participants are mainly motivated by the opportunity to earn money. However, research also suggests that intrinsic motivations play an important role in crowdsourcing (Lakhani, Lohse, Panetta, & Jeppesen, 2007). Such intrinsic, motivational factors are, for instance, testing one's capabilities or seeking feedback from other participants (Lampel et al., 2012).

Hence, individuals in crowdsourcing initiatives are motived by extrinsic and intrinsic factors. Leimeister and colleagues (2009) argue that crowdsourcing activities are comparable

[3] http://www.istockphoto.com/

to sports competitions in which athletes are also motivated by both, extrinsic and intrinsic factors. Intrinsic factors in sports include the winning of trophies, the fun of participating or the feedback of spectators or the media.

2.2.2 Types

The literature identified three ways that firms can use to integrate external individuals in the innovation process: “open call”, “selective open call” and “open search” (Diener & Piller, 2013; Piller & West, 2014). An “open call” or “broadcast search” (Jeppesen & Lakhani, 2010) approach spreads a certain task to a wide, heterogeneous network of external actors. Thus, there is no restriction which individuals may participate (Afuah & Tucci, 2012; Howe, 2006; Piller & West, 2014).

By contrast, the crowdsourcing types “selective open call” and “open search” restrict the participation: In the selective open call, firms identify at first a specific segment of individuals - based on certain characteristics - and then limit participation to this selection (Diener & Piller, 2013; Piller & West, 2014). For instance, only individuals might be allowed to participate who have a high level of expertise is a specific field.

In the “open search” crowdsourcing approach firms select and invite a few users to co-create a service or product in collaboration with the firm. This higher level of collaboration with selected external experts is known as the lead user theory which was developed and widely researched by von Hippel (1986).

This paper will focus on the open call approach, and specifically analyse the subtype of tournament-based crowdsourcing, in that firms attain external knowledge of the crowd in form of a contest. Other subtypes of open call crowdsourcing are idea- or voting contests and internet toolkits which all differ regarding the level of knowledge exchange and regarding the level stakeholders are involved (Franke & Piller, 2004; Leimeister et al., 2009; Lüttgens et al., 2014).

2.3 Crowdsourcing Contests

2.3.1 Overview

Sourcing external knowledge by organizing a tournament (tournament-based crowdsourcing), had already been used many years ago, as discussed earlier. Over the last years, firms started to use crowdsourcing contests as an ongoing business model (Boudreau & Hagiu, 2009) and they evolved from creating product concepts towards a solid way of problem solving for companies (Terwiesch & Xu, 2008).

Tournaments and evaluation methods of performances were discussed in the economics literature drawing from areas such as decision making in politics, sales or sports (Casas-Arce & Martinez-Jerez, 2009; Lazear & Rosen, 1981). In general, two different types of contests exist: the first type contains two firms competing in a two-stage game against each other by first putting efforts into R&D and then bringing the product to the market (Ziss, 1994). Crowdsourcing contests, however, represent the second type of contests since, normally, they are designed as competitive games, in which solvers invest effort and resources in order to win an award (Fullerton & McAfee, 1999).

Moreover, crowdsourcing contests differ from several other contest types by the desired outcome of the firm. Although there may be contests in which several solutions are of interest for the seeking firm, usually only the maximum (best) solution matters in order to solve a certain innovation problem (Terwiesch & Xu, 2008). Thus, a seeker of an innvoation contest will prefer to get one outstanding submission and 100 bad ones, instead of getting 101 good solutions. By contrast, in other types of contests (e.g. salesforce contests), the seeker is keen on maximising the sum of all performances and hence prefers to achieve 101 good performances compared to one outstanding performance but 100 poor other ones (Terwiesch & Xu, 2008).

Girotra, Terwiesch and Ulrich (2010) emphasize the importance of the maximum solution and determine the performance in a crowdsourcing contest by four factors: the number of solutions, solutions' average quality, solutions' variance and finally the best solution (see Figure 3).

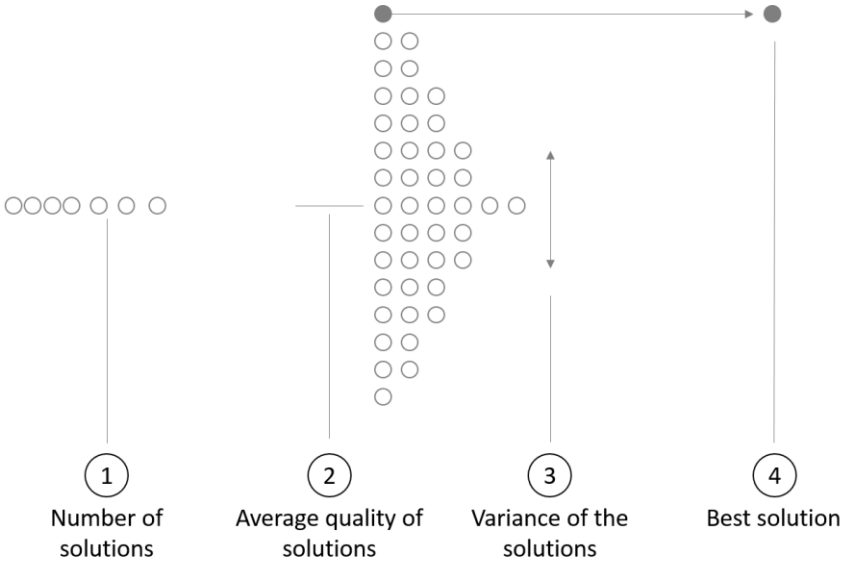


Figure 3: Four Factors determining the Performance of a Crowdsourcing Contest (Adapted from Girotra et al., 2010)

Although crowdsourcing contests evolved quickly and gained popularity over the last years, research on this type of crowdsourcing isn't as advanced (Lampel et al., 2012). The non-uniform terminology might be one reason since the literature refers to "crowdsourcing contests" (Archak, 2010; Zheng, Li, & Hou, 2011) also as "innovation contests" (Boudreau et al., 2011; Terwiesch & Xu, 2008), "design competitions" (Lampel et al., 2012), "crowdsourcing tournaments" (Afuah & Tucci, 2012) or "technology contests" (C. Cohen, Kaplan, & Sela, 2008).

2.3.2 Process

In a crowdsourcing contest, the seeker (firm) broadcasts a problem, including the requirements concerning the submissions, to the audience of potential solvers on a chosen crowdsourcing platform, which can be a firm's own platform or an intermediary platform, such as InnoCentive or TopCoder (Piller & West, 2014). The "request for proposals" explains the problem that a firm is facing. Thereupon, interested solvers screen the request for proposals and decide whether to invest efforts to elaborate and deliver a solution. Contest periods normally range from a very short duration of some hours to contests which are open for a few months (Adamczyk et al., 2012; Boudreau, Lakhani, & Lacetera, 2008; Bullinger & Möslein, 2010). When the contest is over, the seeking firm or a jury of experts screens the submitted solutions and determines which one meets best the communicated requirements (Afuah & Tucci, 2012; Lüttgens et al., 2014). Additionally, the

jury can also involve other solvers of the platform (Haller, Bullinger, & Möslein, 2011). Finally, the winner(s) of the contest receive an award, usually in a monetary form (Leimeister et al., 2009).

2.3.3 Types

Terwiesch and Xu (2008) enhanced the research on crowdsourcing contests by classifying projects (problems) into three different categories: *expertise-based projects*, *ideation projects* and *trial-and-error projects* (see Figure 4). The three problem categories can be distinguished by the level of technical uncertainty and market uncertainty. Technical uncertainty defines to which extent contest problems have a clear solution landscape and whether solvers can anticipate a performance before actually conducting an experiment. Market uncertainty, on the other hand, describes to which extent contest goals and expectations are clearly formulated wherefore a solver is able to anticipate whether the seeker will like a solution (Terwiesch & Xu, 2008).

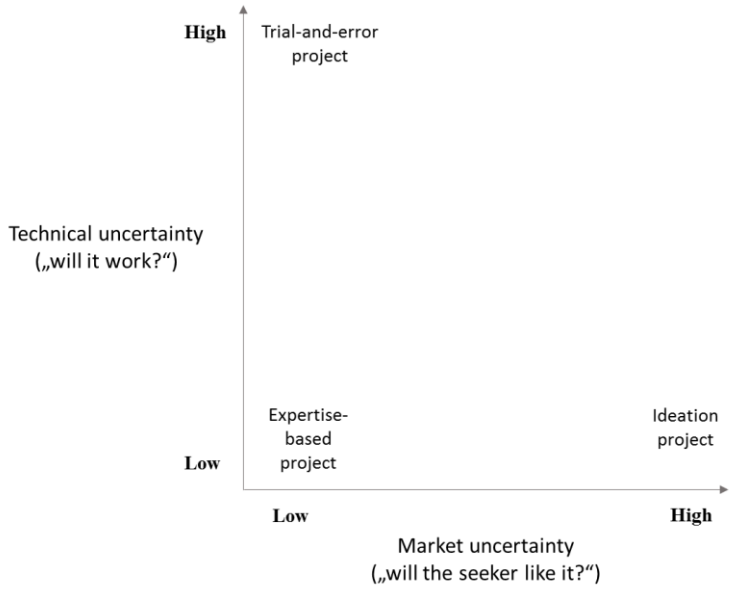


Figure 4: Project Types in Respect to Market Uncertainty and technical Uncertainty (Terwiesch & Xu, 2008)

In *trial-and-error projects*, a solver's performance is mainly determined by the number of experiments, of which the solver will then choose the best performing one. For example, in trial-and-error contests, solvers might be asked to develop a pill which reduces grey hairs. In this project type, solvers are aware of the seeker's expectations (low market

uncertainty) but don't exactly know how to improve a solution (high technical uncertainty) (Terwiesch & Xu, 2008).

In *ideation projects*, the performance is highly determined by the effort a solver puts into his solution. Contests of this type are characterized by a high market uncertainty since the seeker's taste is highly subjective. Typically, such contests are design-related, for instance logo contests on the platform LogoMyWay[4], where firms can ask users to create a business logo. Interestingly, one research paper analysed such ideation contests and found that seekers are more likely to choose a submission as the best one, if the solver is culturally close, thus has a similar taste for design (Bockstedt, Druehl, & Mishra, 2015).

Finally, *expertise-based projects* reflect the last project type. This type is analysed most commonly in prior research and will be also subject of this study. Expertise-based projects are usually engineering problems, for example in the field of software development. They have a low market uncertainty since the expectations and goals of the seeker are clear to the solvers of a contest. In a software engineering contest, for example, solvers know exactly how the requested software solution should work and look like. As the term of the project type implicates, a solver's performance is primarily determined by his expertise (Girotra et al., 2010; Terwiesch & Ulrich, 2009; Terwiesch & Xu, 2008).

Therefore, a higher expertise clearly increases a solver's likelihood of winning an expertise-based contest. The literature suggests that a solver's expertise is a measurement of the solver's past performances in contests (Snir & Hitt, 2003; Terwiesch & Xu, 2008). Research on users from the Chinese crowdsourcing platform TaskCN[5] confirms this and further shows that a user's past performance is a good predictor for his future chances of success (Y. Yang, Chen, & Banker, 2010). Moreover, an experiment revealed that a very small core group of solvers submit nearly 20% of the winning solutions (J. Yang, Adamic, & Ackerman, 2008).

The categorization of projects into three categories, defined by Terwiesch and Xu (2008), has been adopted by several following scientific papers (Boudreau et al., 2008; Yang Yang et al., 2009). Emphasizing the differences between the three problem types is crucial since contest elements, as the number of number of solvers, the award and the problem description, have wide impacts across each type.

[4] http://www.logomyway.com/

[5] http://www.taskcn.com/

2.4 Contest Design Elements

2.4.1 Overview

Extant research analysed crowdsourcing contests from five main perspectives. The economic perspective looks at the ideal design of contest elements, such as the number of solvers, the award structure, the contest presentation, the target group and the contest duration (Adamczyk et al., 2012).

2.4.2 Open versus Limited Participation

One central discussion in prior research is the question whether participation in contests shall be open or limited to a certain number of individuals. Two theories, the *competition effect* and the *parallel path effect* argue for different outcomes.

The *competition effect* states that a limited participation in a contest is preferable for the seeker. The theory explains that a limited number of solvers increases the solvers' efforts since they perceive their likelihood of winning as higher. The other way round, an increase in the number of solvers will lead to reduced efforts by solvers due to a lower perceived likelihood of winning. Consequently, the average quality of all submissions will decrease with more participants (Che & Gale, 2003; Taylor, 1995). The average quality of submissions reflects the mean performance of all submissions which were delivered by solvers. Garcia and Tor (2009) proved the effect in a field experiment with test-takers in a university contest who lowered their efforts when they believed to be competing with a higher number of contestants. In a recent research paper, Boudreau, Lakhani and Menietti (2016) found that solvers with diverse levels of expertise are affected to a different extent by the competition effect. Whereas most participants react negatively – with lowered efforts – to increased competition, high-skilled solvers on the other hand respond positively to competing with more contestants.

In order to mitigate the competition effect of reduced efforts of solvers, a limit regarding participants is preferable. Fullerton and McAfee (1999) even suggest that the ideal number of solvers in contests is in fact just two.

Based on these observations, I propose that:

> **Hypothesis 1a:** *The greater the number of registered solvers, the lower is the average performance of a contest.*

In contrast, the *parallel path effect* (or parallel search effect), argues that an open participation in a contest is preferable for the seeker since an increased number of participants leads to a better maximum performance. The effect considers submitted solutions as parallel experiments and argues that a higher number of submissions increases the variance of solutions in terms of quality. Consequently, the maximum performance, as the extreme value, improves through a higher number of solvers. Since this maximum performance is the only solution a seeking firm is interested in, inferior solutions of other solvers, which result from a higher number of participants, are negligible. Thus, the parallel path effect only focuses on the extreme value in a contest, which improves throughout a greater number of participants. (Boudreau et al., 2008; Dahan & Mendelson, 2001; Nelson, 1961).

Terwiesch and Xu (2008) were the first researchers who combined both theories, the parallel path and the competition effect, in a theoretical model which examines which effect has a stronger impact. They argue that the benefits of the parallel path effect can outweigh the disadvantages of a lower solution equilibrium, caused by the competition effect. Hence, according to the researchers, a higher number of solvers is preferable for the seeking firm since the maximum solution improves, in quality, although the average of submitted solutions decreases.

Accordingly, I state the following:

> **Hypothesis 1b:** *The greater the number of registered solvers, the higher is the maximum performance of a contest.*

Literature in the field of open innovation discusses the openness of a firm in relationship to its innovation performance (Laursen & Salter, 2006; Salter, Ter Wal, Criscuolo, & Alexy, 2014). It was found that an increasing number of external knowledge sources (increasing openness) improves the innovation performance of a firm since the variety of external knowledge leads to better solutions. Therefore, an increasing number of knowledge sources raises the innovation performance of the firm. However, this increase only holds until a certain threshold. Beyond that point, benefits of more external knowledge sources are outweighed by a nonlinear increase of costs, associated to more knowledge sources. Thus, the authors investigated an inverted u-shaped relationship between the number of external knowledge sources and the firm's innovation performance and found evidence of diminishing returns when too many external knowledge sources are involved (Laursen & Salter, 2006).

Crowdsourcing contests operate in a similar environment, also being confronted with the question how 'open' a contest should be designed – hence how many solvers should ideally participate to maximize the contest's outcome. The level of openness therefore reflects the number of solvers that are participating in a crowdsourcing contest. The outcome variable of Laursen and Salter (2006) is defined as the innovation performance of a firm. In the context of crowdsourcing contests, the outcome of a contest is represented by the best solution that the firm is able to attain.

As presented, Terwiesch and Xu's (2008) model argues that the parallel path effect outweighs the competition effect. However, it is not further discussed whether the relationship of participants and the contest performance might be curvilinear, as it was found in the open innovation literature (Laursen & Salter, 2006; Salter et al., 2014). In the context of crowdsourcing contests, a curvilinear relationship may be in place, due to heavily decreasing efforts by solvers when competing with a very high number of participants. Moreover, solvers can choose their contest of interest from a large offer of available contests (Afuah & Tucci, 2012). Hence, when being confronted with a very high level of competition, solvers are might shift their efforts to a different available contest with fewer registered solvers, since winning chances are higher. Thus, the maximum performance of the contest is expected to decrease beyond a certain number of solvers.

Based on these observations, I propose that:

> **Hypothesis 1c:** *The number of participants is curvilinearly related (takes an inverted U-shape) to the maximum performance of a contest.*

2.4.3 Monetary Award

As discussed earlier, monetary awards are part and parcel of crowdsourcing contests. Thus, the optimal reward structure is a major field of research. It discusses, for instance, whether multiple, proportional awards or winner-takes-all contest designs are preferable. In general, research suggests that multiple prices enhance participation because solvers perceive the likelihood of winning a price as higher (Cason et al., 2010).

In expertise-based contests - in which the solver's expertise highly correlates with the likelihood of winning a contest - winner-takes-all awards offer stronger incentives to highly skilled solvers. Participants with lower skills, on the other hand, prefer contests

with multiple awards since their perceived chances of winning the first price are lowered with a higher number of solvers (Terwiesch & Xu, 2008).

Research also showed that higher contest awards capture more solvers (Lazear & Rosen, 1981; Snir & Hitt, 2003). This is caused by the fact that a higher reward offers a better compensation for the cost of participation - the time costs each solver invests to grasp and solve the problem (Shao et al., 2012).

Furthermore, research on crowdsourcing contests showed that the award value has a positive impact on the solvers' performances in a contest (Archak, 2010; Boudreau et al., 2008). Thus, the greater the reward, the higher the efforts to solve a problem, leading to better performances and solutions.

As the only study, Bockstedt and colleagues (2015) considered diversity among contestants, in terms of their national wealth, in respect to the monetary award. Since crowdsourcing contests are held online and are accessible for users from all over the world, monetary values, as a motivation for contest winners, vary in their attractiveness to solvers from different countries. The researchers showed that national wealth influences the level of effort that contestants invest to solve a problem. Accordingly, solvers from countries with a lower level of GDPP (gross domestic product per capita) invest more effort than other users to solve a problem, since the monetary award is relatively more attractive for them.

According to the literature, the monetary award can positively influence a contest's participation and performance, wherefore I hypothesize that:

Hypothesis 2a: *The greater the monetary award, the higher the level of participation.*

Hypothesis 2b: *The greater the monetary award, the higher the maximum performance of a contest.*

2.4.4 Problem Presentation

Firms broadcast their problems, including the requirements for potential solutions, in a "request for proposals". However, research about the impact of the problem presentation (or problem description) on the solvers' participation and performance in crowdsourcing contests is scare.

Problem statements have to be written in a clear and simple manner so that solvers are able to understand the problem's requirements (Afuah & Tucci, 2012; Piller & West,

2014; Spradlin, 2012). This aspect is eminently important because winning submissions in crowdsourcing contests are often submitted by solvers who have expertise in a field which is vastly different to the expertise field of the problem (Jeppesen & Lakhani, 2010). Research on InnoCentive contests revealed, for instance, that a solver with expertise in a totally different field was able to solve a toxicology problem that the seeking pharmaceutical firm had previously not been able to solve (Lakhani & Jeppesen, 2007).

Lazear and Rosen (1981) showed that higher awards in contests attract more solvers. This is caused since the compensation for learning costs - that each solver has to invest in order to understand the problem requirements - is higher (Shao et al., 2012). Those costs increase further where solvers require more time to grasp the problem statement. According to the switching cost theory, a consumer chooses the option with the lowest switching costs when establishing a new relationship (Klemperer, 1995). In the context of crowdsourcing contests, solvers will therefore participate more likely in contests with low learning costs, meaning low switching costs (Yang Yang et al., 2009). Hence, solvers are more likely to participate in contests with low switching costs, meaning that it takes fewer time to read and understand the problem description.

Prior empirical experiments on crowdsourcing contests considered the description length as the only variable for analysis. Other research in the related field of crowdfunding additionally analysed other attributes such as the readability of the description text and the number of URLs within the description. They found that crowdfunding projects are more likely to be successful when a text is written in a more sophisticated manner, and when URL links, providing additional information, are present in the text (Greenberg et al., 2013; Xu et al., 2014). This type of analysis is obvious to also conduct in the neighboured field of crowdsourcing contests.

Yang and colleagues (2009) showed that crowdsourcing contests on the Chinese platform TaskCN were able to capture more solvers, if a problem description was short. The authors analysed two different types of contests since ideation projects (e.g. creative writing) but also expertise-based projects (e.g. software development) were examined. However, the impact of the text length on solvers' participation might differ by contest type, since the competition effect is unequally strong in those two contest types (Boudreau et al., 2011). This study aims to investigate the impact of the description text length explicitly for expertise-based crowdsourcing contests.

Following these observations, I propose that:

Hypothesis 3a: *A problem description with lower related learning costs positively affects participation in a contest.*

Archak (2010) investigated the impact of the problem description length on the solver's performance but couldn't find any significant impact. It needs to be stressed that this study analysed the performance of all participating solvers. Due to a lack of research, it is unclear if a longer description - which implies a higher level of specificity – affects the maximum solution in a contest. The problem description length might have a different impact on the average and the maximum performance on a contest.

Research shows that the quality of crowdsourced solutions can be unsatisfactory when problem descriptions do not contain all needed problem-specific requirements (Eric von Hippel & von Krogh, 2013). Therefore, the problem description has to have a high level of specificity in order to keep the scope of interpretation for solvers as limited as possible (Lüttgens et al., 2014). This is particularly crucial in expertise-based contests where seekers have very clear product expectations and don't aim to source creative and novel ideas.

Following these observations, I propose that:

Hypothesis 3b: *A problem description with a higher level of specificity positively affects the maximum performance in a contest.*

3 RESEARCH METHODOLOGY

3.1 Data

In order to test the impact of contest elements on the contest performance, a data set from the crowdsourcing contest platform TopCoder (TC) was examined. TC was established in 2001 and is nowadays the leading platform in the field of software engineering contests, having nearly one million active solvers (Topcoder, n.d.-a). Seekers on the platform are Fortune 1000 companies such as AOL, Best Buy or Eli Lilly (Boudreau et al., 2011, 2016).

TC uses an open call approach, meaning that any interested user can sign up to take part in the contest for free. Participants are only asked during the registration to enter a few personal information, including their e-mail address and name. Furthermore, they can specify, on a voluntary basis, in which types of contests they are interested in and define their skills regarding the various technologies and programming languages. Solvers on TC work alone, but can view on the contest's homepage which other solvers signed up and how high their expertise scores are.

The jury board, reviewing submissions, consists of community members who fulfil certain criteria and get trained for this task. Some review boards also include project staff and/or employees of the seeking firm (Topcoder, n.d.-b).

In contrast to other similar platforms, such as InnoCentive, TC provides the data of participants including their performances for current, but also for past contests. This enables to conduct empirical analyses regarding solvers' performances. TC also launches design and data science contests – however, access to these tournaments is not fully public and the review process is different. Therefore, only contests from the software engineering category were analysed. These contests can be classified as expertise-based contests.

In order to attain the data set, the TC website was crawled with the tool Kimono[6], which allows to fetch chosen website elements automatically from a structured website. As a result, a set of 1,109 contests (781 *Assembly* contests, 328 *UI Prototype* contests) was downloaded. The contest categories (*Assembly* and *UI Prototype*) were chosen since they - unlike other categories - show each solver's expertise score at the moment of a contest. This is crucial in order to control for the expertise in the regression models.

[6] www.kimonolabs.com

The sample included all existing contests from the two categories in the time until 29th February 2016. General contest attributes, such as the duration in days, the monetary award, the problem description and the status (successful/ unsuccessful) were saved in a database. Additionally, subpages of the contest, including information about participating and submitting solvers, were crawled. Herewith, data of 25,584 registered solvers (1,933 of them delivered a solution) were saved and matched with the general contest attributes.

Reviewing the data set, a clear decline of the number of contests could be assessed since the end of 2014 (see Figure 5). However, since not all contest categories were included the sample, a general negative development of the TC platform cannot be claimed.

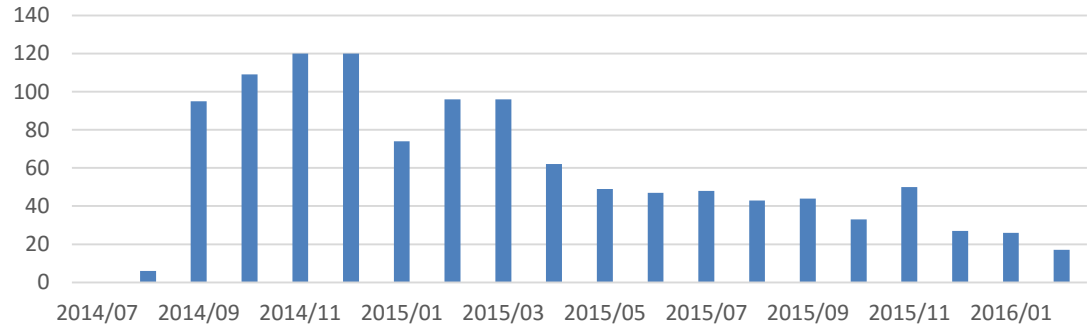


Figure 5: Number of Contests in the Sample (Categories: Assembly and UI Prototype) by Month (Source: own work)

In order to create the final data set, several contests were excluded from the sample of 1,109 contests. Unsuccessful contests (n = 194) were removed from the sample since no contest performance is measured in these contests. Contests fail when no solutions are submitted, the seeker decides to cancel the contest or for other unspecified reasons. Besides excluding these unsuccessful contests, two other contests were removed since one contest was a promotion, hence not a serious contest while the second contest's subpages were technically not fully accessible. After running these data cleaning steps, the final sample included 913 contests.

The sample was analysed with the statistics software R-Studio[7] and the additional package stargazer (Hlavac, 2015). The software combination allows to run and present regression models and graphical data analyses.

[7] https://www.rstudio.com/

3.2 Variables

The analysis exploits data downloaded directly from the TC webpage. The variables considered for analyses are presented in Table 1. All of these variables are also selected in similar studies (Archak, 2010; Boudreau et al., 2011, 2008; Shao et al., 2012; Snir & Hitt, 2003; Yang Yang et al., 2009). They can be clustered into five categories*: Contest performance* variables reflecting the achieved scores, general *contests characteristics* variables such as the contest duration, *solvers' characteristics* variables describing solvers' expertise*, problem presentation* variables analysing the problem description and finally the *platform characteristics* variables.

Table 1: Analysed Variables

Variable	Definition
Contest Performance	
Maximum Score (MS)	The highest score in a contest.
Average Score (AS)	The average of all scores in a contest.
Contest Characteristics	
Number of Registered Solvers (NRS)	Number of solvers who signed up for a contest.
Contest Duration (CD)	Duration of a contest in days.
Monetary Award (MA)	Total monetary amount paid out to the winner of a contest.
Skill Variety (SV)	The number of skills (e.g. programming languages) which are required to solve the problem; Defined by the seeker.
Difficulty (D)	Difficulty of a posted problem. Calculated by the percentage of solvers who finally also submitted a solution (Completion Rate).
Solvers' Characteristics	
Expertise Maximum (EM)	Highest expertise ranking among all solvers in a contest.
Expertise Average of Registered Solvers (ERS)	The average of expertise rankings of participating solvers in a contest.
Expertise Average of Solvers with Submissions (ERSS)	The average of expertise rankings of solvers who finally submitted a solution in a contest.
Problem Presentation	
Description Length (DL)	The text length of the problem description.
Description Readability (DR)	The readability of a problem description measured with the ARI (Automated Readability Index).
Number of Links in the Description (NLD)	The number of outgoing URLs in the problem description.
Platform Characteristics	
Number of Open Contests (NOC)	The number of other contests which are running at the same time on TC.
Platform Maturity (PM)	The number of days since the TopCoder platform was launched.

Source: Own work

3.2.1 Contest Performance

In order to test whether certain contest attributes have an impact on the contest performance, the maximum score and average score are analysed in separate models. Performance scores on TC are rated by a jury of 2-3 reviewers who evaluate submissions based on scorecards (see Appendix H). These scorecards determine to which extent seeker's requirements are met and whether the submission is technically feasible. Final scores vary in the range of 0 to 100 whereas 100 is the maximum score that can be achieved (Topcoder, n.d.-b).

The *average score* (AS) simply reflects the sum of all scores, divided by the number of submissions. This figure is calculated in order to test the existence of a competition effect (Hypothesis 1a).

The *maximum score* (MS) reflects the best delivered solution in a contest and is required in order to test whether an increased number of solvers leads to a better maximum performance (Hypothesis 1b and 1c).

3.2.2 Number of Solvers

The *number of registered solvers* (NRS) is the main explanatory variable and describes how many solvers are competing. Solvers have to sign up for a contest in a specific time frame which normally lasts a few days. Only after that period the contest starts and registration is closed. Thus, all solvers exactly know at the time the contest starts against how many contestants they are competing. This procedure is of cardinal importance in order to investigate the impact of participation on solvers' performance.

In the analysed sample, the number of solvers ranges between 1 and 60, the mean accounting to ~23. However, 95% of the contests in this sample have a range between 7 and 44.3 solvers. Figure 6 shows the distribution of the variable in detail.

Besides acting as a control variable in the regression models that analyse the impact of contest attributes on the contest performance, *the number of registered solvers* (NRS) is taken as the outcome variable in another regression model in order to test the impact of contest attributes on the participation in a contest.

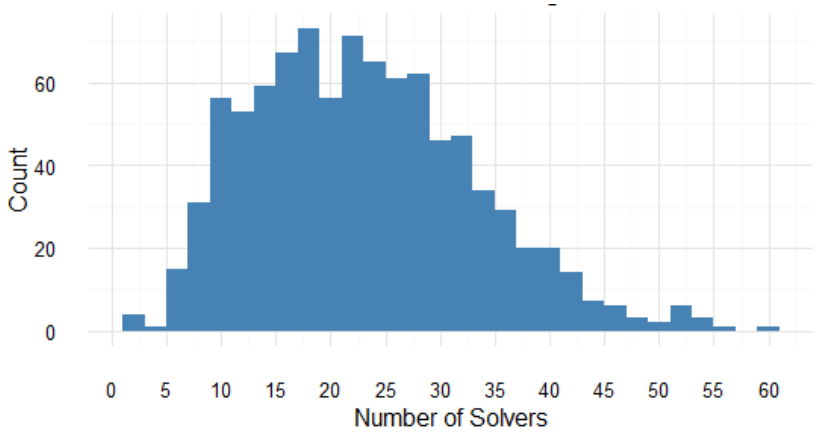


Figure 6: Distribution of the Number of Solvers in a Contest (Source: own work)

3.2.3 Monetary Award

Solvers in crowdsourcing contests are strongly extrinsically motivated (Brabham, 2010; Terwiesch & Xu, 2008) wherefore the reward is considered as a control variable. On TC, the *Monetary Award* (MA) is always paid out to the two best submissions of the contest, whereas the winner gets two third of the total price while the second place gets the remaining third. Since this ratio does not vary and is fixed for all contests, it is sufficient to consider the total monetary award as the control variable. The variable is measured in $1,000.

3.2.4 Problem Presentation

For seekers, the problem statement is the main point of contact with potential solvers. Therefore, it is important that it is well written since potential solvers base their decision, whether to participate in a contest or not, strongly on the quality of the problem description. Thus, writing a problem statement is one of the most important design elements of a contest (Lüttgens et al., 2014).

Following the methodology of Archak (2010) and Yang and colleagues (2009), the *description length* (DL) is used as one variable to represent the problem statement in a contest. While Archak (2010) measured the text length in pages, here, the number of characters in a problem description is counted, following the approach of Yang and colleagues (2009). However, URLs within the text were excluded from the text length and analysed separately. The text length in the sample varies considerably amongst all the contests. The natural log of this variable is used.

Secondly, the variable *description readability* (DR) analyses how easy a problem statement is written. The Automated Readability Index (ARI) (Smith & Senter, 1967) is applied to measure a problem description's readability. Introduced to analyse the readability of technical documents of the US Army, it's suitable to be used for highly technical problem descriptions of TC contests. The index considers the length of words and sentences and then calculates a score whereby a low score means that a text is easy to read while a high score implies a more complicated and specified content. The natural log of this variable is used.

As a third variable in the category of the problem presentation, the *number of links in the description* (NLD) is analysed. Problem descriptions of contests on TC often include links to other homepages where each additional link increases the time each solver has to invest to grasp the problem's requirements.

3.2.5 Control Variables

Since achieved scores and participation in a contest can be correlated to several other attributes of a contest, it is needed to control for the following variables.

Contest Duration (CD): Contests on TC differ in terms of the duration wherefore this variables controls for that variance. It is measured in days and reflects the difference of the start and end date of a contest (Leimeister et al., 2009; Shao et al., 2012; Snir & Hitt, 2003)**.**

Difficulty (D): The difficulty of a problem is a very impactful variable but it is hard to measure. However, this analysis considers the completion rate as the difficulty of a contest. Hence, the difficulty of a contest is defined as the percentage of solvers who have ultimately submitted a solution, divided by the total number of registered solvers. A low ratio means that a contest was difficult since most solvers could not find a solution whereas a high ratio reflects an easier problem. This method is also used by other research in order to control for the difficulty of a contest (Shao et al., 2012; Yang Yang et al., 2009). The natural log of this variable is used.

Skill Variety (SV): A seeker in a contest defines the skills which are required to solve a problem. On TC, these are mostly technologies such as Java, HTML or CSS. This variable represents the number of technologies which are required to solve the problem. It is also taken as a control variable in other experiments (e.g. Archak, 2010).

Expertise: The expertise score indicates a solver's position within the overall skill distribution of members (Boudreau et al., 2011). Since the sample contains expertise-based contests, the solvers' expertise levels are the primarily factor for their performance and therefore important to control for in the models. TC measures a solver's expertise with an Elo-based system, which has its origins in the sport of chess (Van Der Maas & Wagenmakers, 2005). TC only considers the last 50 contest performances of solvers for the calculation of the expertise score (Topcoder, n.d.-c). Here, different expertise variables are considered, following the approach of Boudreau and colleagues (2011): the average of *expertise among registered solvers* (ERS), the *average of expertise of solvers with submissions* (ERSS) and the *maximum expertise* (EM) of all registered solvers.

Number of Open Contests: Since a varying offer of available contests on the TC platform could affect participation, the variable *number of open contests* (NOC) controls for the number of available open contests, at the time when a contest is launched.

Platform Maturity: Analysing online platforms requires controlling for the platform maturity (PM) since network effects can affect the number of users on the platform (Snir & Hitt, 2003). The platform maturity is measured by counting the days from 1st March 2014 until the start date of each contest. The natural log of this variable is used.[8]

3.3 Estimation Approach

In order to test the hypotheses, four regression models are being estimated, using the ordinary least squares (OLS) method, similar to other research experiments in the field of crowdsourcing (Archak, 2010; Boudreau et al., 2011; Shao et al., 2012).

The first model sets the level of participation as the outcome variable. The expertise variables are not present in this model since solvers register for a contest by having incomplete information about other solvers since they might register at a later point at time. Thus, the variables regarding the expertise (EM, ERS, ERSS) are not relevant for this analysis.

Model 1: Number of Registered Solvers (NRS)

$$NRS = \beta_0 + \beta_1 CD + \beta_2 MA + \beta_3 SV + \beta_4 DL + \beta_5 DR + \beta_6 NLD + \beta_7 PM + \beta_8 NOC$$

[8] Alternatively, instead of the platform maturity variable, the models were run with time dummy variables for each month. Results remained unchanged, only showing slightly different coefficients.

The following three models set the two different contest performance indicators as the outcome variable. In order to test the evidence of a competition effect (H1a), Model 2 sets the *average score* (AS) as the outcome variable. Model 3a sets the *maximum score* (MS) as the outcome variable in order to test the parallel path effect in coexistence with the competition effect (H1b). Finally, Model 3b includes a quadratic variable for the number of registered solvers in order to test H1c.

Model 2: Average Score (AS)

$$AS = \beta_0 + \beta_1 NRS + \beta_2 CD + \beta_3 MA + \beta_4 SV + \beta_5 D + \beta_6 EM + \beta_7 ERS + \beta_8 ERSS + \beta_9 DL + \beta_{10} DR + \beta_{11} NLD + \beta_{12} PM$$

Model 3a: Maximum Score (MS)

$$MS = \beta_0 + \beta_1 NRS + \beta_2 CD + \beta_3 MA + \beta_4 SV + \beta_5 D + \beta_6 EM + \beta_7 ERS + \beta_8 ERSS + \beta_9 DL + \beta_{10} DR + \beta_{11} NLD + \beta_{12} PM$$

Model 3b: Maximum Score (MS) (including quadratic factor)

$$MS = \beta_0 + \beta_1 NRS + \beta_2 NRS^2 + \beta_3 CD + \beta_4 MA + \beta_5 SV + \beta_6 D + \beta_7 EM + \beta_8 ERS + \beta_9 ERSS + \beta_{10} DL + \beta_{11} DR + \beta_{12} NLD + \beta_{13} PM$$

4 RESULTS

Table 2: Results of the Linear Regression

	Dependent variable:			
	(NRS)	(AS)	(MS)	
Model	1	2	3.1	3.2
Number of Registered Solvers (NRS)		-0.091***	0.138***	0.311***
		(0.033)	(0.025)	(0.081)
NRS (squared)				-0.003***
				(0.001)
Contest Duration (CD)	1.079***	0.025	-0.050	-0.054
	(0.172)	(0.158)	(0.121)	(0.120)
Monetary Award (MA) (in $1,000)	2.367***	-1.432***	-1.092***	-1.227***
	(0.459)	(0.413)	(0.316)	(0.321)
Skill Variety (SV)	0.268*	-0.152	-0.209**	-0.215**
	(0.142)	(0.125)	(0.096)	(0.095)
Difficulty (D) (log)		-1.488***	3.212***	3.385***
		(0.436)	(0.333)	(0.341)
Expertise Maximum (EM)		-0.001	0.003***	0.002***
		(0.001)	(0.001)	(0.001)
Expertise Average of Registered Solvers (ERS)		0.002	-0.002	-0.002
		(0.002)	(0.002)	(0.002)
Expertise Average of Solvers with Submissions (ERSS)		0.008***	0.006***	0.006***
		(0.001)	(0.0005)	(0.0005)
Description Length (DL) (log)	-0.907*	-3.952***	-2.563***	-2.467***
	(0.467)	(0.421)	(0.321)	(0.323)
Description Readability (DR) (log)	1.339	2.590**	1.964**	2.026**
	(1.362)	(1.193)	(0.911)	(0.910)
Number of Links in the Description (NLD)	-0.249***	0.259***	0.153***	0.161***
	(0.053)	(0.047)	(0.036)	(0.036)
Platform Maturity (PM) (log)	12.059***	0.949	-0.091	-0.313
	(0.749)	(0.753)	(0.575)	(0.582)
Number of Open Contests (NOC)	-0.236***			
	(0.053)			
Constant	-52.216***	103.809***	107.365***	106.687***
	(7.026)	(6.422)	(4.904)	(4.902)
Observations		913		
R^2	0.462	0.352	0.309	0.313
Adjusted R^2	0.457	0.343	0.300	0.303

*, **, and *** indicate statistical significance at the 10%, 5%, and 1% levels, respectively. Robust standard errors in parentheses. Model 3.2 includes the squared NRS variable, whereas 3.1 doesn't. Source: Own work

On the final data set of 913 crowdsourcing contests, linear (OLS) regression models were run on the participation (NRS), the average performance (AS) and the maximum performance (MS). The results can be found in Table 2. The descriptive patterns and correlation matrix are presented in Appendix A and Appendix B, respectively.

For each linear regression model, the f-test statistic was calculated and was found to be significant (at the 1% level). Moreover, each model was checked for heteroscedasticity with the Breusch Pagan test (Breusch & Pagan, 1979), resulting in p-values smaller than 0.01. The variance inflation factors (VIF) are presented in Appendix C, showing that the three basic regression models are free of multicollinearity (maximum variance inflator factor = 2.426). Only the last model, which adds the squared NRS variable shows naturally a higher variance inflation factor for this variable (24.728).

Furthermore, the regression model 1, which analyses the impact of contest elements on the level of participation, was additionally run on an alternative data sample, which included - besides successful contests - also failed contests. This analysis was conducted in order to investigate potential differences, when these unsuccessful contests are added. Potentially contests might have failed due to a poor problem description. However, the coefficients and significance levels showed the same results as in the model analysing the final data set.

4.1 Number of Solvers

In this section, the relationship between the level of competition and the contest performance is examined. Both opposing theories, the competition effect and the parallel path effect, as well as a potential curvilinear relationship are being investigated.

In terms of descriptive patterns, the *number of registered solvers* (NRS) ranges between 1 and 60 (Mean: 23.46) while the *average score* (AS) and the *maximum score* (MS) variables have a mean of 88.65 and 92.62, respectively (see Appendix A).

The first point of discussion is the impact of the *number registered solvers* (NRS) on the average performance in a crowdsourcing contest. Following Hypothesis 1a, a negative coefficient on the *number of registered solvers* (NRS) is expected. The results, presented in Table 2 (Model 2), show indeed that a higher number of solvers results in a reduced average performance ($ß = -0.091$, $p < 0.01$). These results confirm H1a, aligned with the view of the competition effect.

Subsequently, the evidence of the parallel path effect is investigated. According to this theory, the *maximum score* (MS) in a contest should increase with a *higher number of solvers* (NRS). Hypothesis 1b therefore suggests a positive effect of NRS on the maximum performance in a contest. The results (Model 3.1) confirm H1b ($ß = 0.138$, $p < 0.01$).

Finally, a third analysis investigated whether a curvilinear relationship between the *number of registered solvers* (NRS) and the *maximum score* (MS) is existent. As shown (Hypothesis 1b), analysing the maximum performance, outweigh the advantages of an increased number of solvers the disadvantage of a lowered performance average, caused by the competition effect (Hypothesis 1a). However, hypothesis 1c proposes that this relationship changes at a certain number of participants and gets negative, thus a curvilinear relationship is in place. The results (Model 3.2) also confirm this hypothesis and indicate a negative u-shaped relationship between the two variables (NRS: $ß = 0.311$, $p < 0.01$; NRS squared: $ß = -0.003$, $p < 0.05$). Hence, adding competitors to a contest leads to a better contest performance. However, this positive effect declines gradually and finally even a negative return sets in, wherefore adding further competitors beyond this threshold let the maximum performance in a contest decline. Nevertheless, it needs to be stressed that the coefficient (NRS squared), indicating a negative curvilinear relationship, is very small, wherefore the impact is rather low.

By adding the quadratic variable to the model, the explanatory power (Adjusted R2) slightly increases from 30.0 (Model 3.1) to 30.3 (Model 3.2).

4.2 Monetary Award

In this section, the impact of the monetary award on the level of participation and the maximum performance is examined.

In total, the rewards of the contests in the sample amounted to $2.2 million (see Table 3). The *monetary award* (MA) for the analysed TopCoder contests ranges between $300 and $5,250 (Mean: $1,958.27) (see Appendix A).

Table 3: Total Rewards of Contests in the Sample

	Failed Contests	Successful Contests	Total
Monetary Award	$ 413.975,00	$ 1.787.900,00	$ 2.201.875,00

The first analysis regarding the award in a contest examines the impact of an increased award on the *number of registered solvers* (NRS). It was expected that a higher *monetary award* (MA) positively affects the *number of registered solvers* (NRS) (H2a). The finding (Model 1) confirms this hypothesis (ß = 2.367, $p < 0.01$).

Hypothesis 2b states that an increased *monetary award* (MA) results in an improved *maximum score* (MS) since solvers are extrinsically motivated and therefore invest more efforts into their solutions. The results (Model 3.1 and Model 3.2), however, show the opposite, reflected by a negative coefficient (Model 3.1: ß = -1.092, $p < 0.01$; Model 3.2: ß = -1.227, $p < 0.01$). Hence, higher price awards lead to a lower maximum performance, which is contrary to theory and previous experiments. Moreover, this impact is also present and significant regarding the average performance in a contest (Model 2), wherefore it can be stated that all performances lower when the award is increasing.

Thus, the monetary award, as an extrinsic motivation, enhances the level of participation but worsens the solvers' performances.

4.3 Problem Presentation

The last part of analysis investigates the impact of the problem description on the level of participation and the performance. Specifically, the text length, the text readability and the number of links in the problem statement are analysed.

The *description readability* (DR) value reflects the Automated Readability Index (ARI) (Smith & Senter, 1967). A low score reflects short words and sentences, wherefore a text is easy to read. A high score implies a more complicated and specified content.

Hypothesis 3a suggested that a problem statement with lower learning costs, meaning that it takes less time to fully understand the task, leads to an increased number of contestants. Hence, learning costs increase when the values for the *description length* (DL), the *description readability* (DR) and the *number of links in the description* (NLD) are higher, because solvers have to invest higher efforts to fully grasp the problem description.

Results partly confirm this hypothesis (Model 1): The *description length* (DL) (ß = -0.907, $p < 0.1$) and the *number of links in the description* (NLD) (ß = -0.236, $p < 0.01$) negatively affect, as suggested, the *number of registered solvers* (NRS). Thus, the shorter the text and the fewer URL links in the text, the higher is the number of participants. The

description readability (DR) has, contrary to the hypothesis, a positive coefficient (ß = 1.339). However, this finding is statistically insignificant.

H3b explores the impact of the problem presentation on the maximum performance. It was expected that an extensive and detailed problem statement improves the maximum solution, since the scope of interpretation regarding the solution's requirements is smaller (Hypothesis 3b). Hence, solvers are clearer about the seeker's requirements and can meet the expectations easier. This hypothesis is partly confirmed by the regression models (Model 3.1 and Model 3.2). As suggested, the *number of links in the description* (NLD) (Model 3.1: ß = 0.153, $p < 0.01$; Model 3.2: ß = 0.161, $p < 0.01$) and the *description readability* (DR) (Model 3.1: ß = 1.964, $p < 0.05$; Model 3.2: ß = 2.026, $p < 0.05$) increase the *maximum score* (MS) in a contest. Against expectations, the *description length* (DL) has an opposite effect since the maximum performance decreases with a longer text (Model 3.1: ß = -2.563, $p < 0.01$; Model 3.2: ß = -2.467, $p < 0.01$). This result is surprising since it was expected that a longer text conveys more information which reduces information asymmetry between seeker and solver.

5 DISCUSSION

The purpose of this study was to examine how certain contest attributes in crowdsourcing contests should be ideally designed in order to maximize the outcome of a contest for the seeker (the firm). This aspect of research on crowdsourcing contests was incomplete and required further investigation (Adamczyk et al., 2012). Closing this gap of research, this study particularly discussed three design elements of a contest. The monetary award, the problem description and the number of participants, each being examined regarding its impact on the level of participation and the performance of a contest. Thus, this study aimed to find empirical evidence on the ideal contest design, based on a large and current data set of 913 contests from the crowdsourcing platform TopCoder.

Other major studies mostly investigated contests which were launched several years ago. For example, Boudreau and colleagues (2011) analysed contests that took place between 2001 and 2007. However, firms' usage of crowdsourcing contests clearly changed over the last decade (Boudreau & Hagiu, 2009). Therefore, the sample of contests in my study is able to provide a more relevant analysis of firm's current usage of crowdsourcing contests to source innovations. Moreover, the large sample size itself elevates the relevance of this study.

Summarizing the findings of this study, it was shown that more solvers in a contest lead to a lower average performance but a higher maximum performance, wherefore it is preferable for a firm to have many participants competing. However, the benefit of adding solvers gradually declines and finally reverses and negative returns set in. Hence, an inverted u-shape relationship between the number of solvers and the maximum performance was found in this study.

Moreover, findings revealed that a higher monetary award motivates more solvers to participate in a contest, but, surprisingly, decreases the solvers' performances.

Finally, the problem description was examined and the findings show that a shorter problem description results in a greater number of participants and a better maximum contest outcome. Moreover, a problem description which contain URL links, directing the solver to additional information, declines the level of participation but improves the maximum performance. Lastly, findings further show that a problem description with a high readability score, reflecting a detailed and sophisticated text, leads to a better maximum performance.

5.1 Theoretical Implications

This study provides some theoretical implications. Firstly, it enriches the academic discussion regarding the ideal number of solvers in a contest. Although the term crowdsourcing implies that a problem is broadcasted to a large crowd of people, the literature is not consent whether such an unrestricted participation model is actually ideal regarding the contest outcome. Two contrary theories, the competition effect and the parallel path effect, were presented and their existences finally empirically tested. It was given evidence for the theory of the competition effect, suggesting a lowered average performance when having a higher number of contestants (Fullerton & McAfee, 1999; Garcia & Tor, 2009). Hence, the expected value of all solutions decreases with a higher level of participation.

The parallel path effect, as an opposing theory, argues for an improving maximum solution when solvers are being added to a contest (Boudreau et al., 2008; Dahan & Mendelson, 2001; Nelson, 1961). The effect was tested in a further hypothesis and was also proven to be existent. Thus, as Terwiesch and Xu (2008) suggested in their theoretic model, the parallel path effect is able to outweigh the competition effect. This means that a higher number of contestants in a tournament is preferable for a seeking firm since it improves the best delivered solution.

These findings are impactful for the theoretic discussions regarding the ideal number of participants in crowdsourcing activities. Firstly, and against the conclusions of many economists (e.g. Fullerton & McAfee, 1999), who argued that participation in contests should be restricted to two solvers, the results of this analysis show a greater number of solvers is desirable for a firm. Thus, confirming empirically the theoretic model of Terwiesch and Xu (2008), the competition effect and the parallel path effect coexist but, in the view of a firm, the benefits of the parallel path effect outweigh the negative impact of the competition. It is important to note that this finding only applies to contests where only the best outcome (best solution) is of interest for the firm. This is the case for innovation problems (Girotra et al., 2010), which were the subject of this analysis.

Furthermore, a possible curvilinear relationship between the maximum performance and the number of solvers in a contest was examined. Prior research in the field of crowdsourcing had not explored this possibility wherefore the analysis clearly enriches the academic discussion regarding the ideal number of contestants. It was expected that, adding contestants beyond a certain level would have a detrimental effect on the maxi-

mum performance, since the correlation effect exponentially increased and finally outperforms the parallel path effect. Thus, a negative u-shaped relationship between the number of solvers and the maximum performance was assumed. The results of the regression model confirmed this hypothesis.

This finding is impactful for academic research, since it gives evidence for the existence of a curvilinear effect, which means that the parallel path effect does not indefinitely outweigh the competition effect but reverses at a certain number of contestants. Hence, this study was able to apply the analysis of Laursen and Salter (2006) in the field of open innovation to the question how many solvers should ideally participate in a crowdsourcing contest. As in Laursen and Salter's study, it was shown that too many external sources (solvers) are not beneficial for the firm. Hence, the finding of a curvilinear effect advances the theoretic research on crowdsourcing contests.

Furthermore, this study shed light on the role of a problem description in a crowdsourcing contest. The only major scientific experiment which analysed the problem statement of a crowdsourcing contest, is the one from Yang and colleagues (2009) which found that shorter problem descriptions capture more solvers. However, the analysed contests of the study varied strongly in terms of award value and task complexity and furthermore included expertise-based projects but also ideation projects. Thus, it is expected that the results of my study provide more relevant and clear implications for research. It was shown that in expertise-based projects, a shorter problem description improves both, the maximum performance and the level of participation in a contest.

As the first empirical study in the field of crowdsourcing contests, the text readability and the number of links in a problem description, as further design elements, were examined in regards to their impacts on a contest's performance and participation. Therefore, the study provides theoretical implications since no prior research had explored these factors in the field of crowdsourcing contests yet.

By applying the switching cost theory to crowdsourcing contests, it was shown that solvers are more likely to participate in a contest if its switching costs are low. Hence, solvers prefer a contest with a shorter problem description and fewer inserted URL links since they need fewer time to grasp the problem's requirements.

5.2 Managerial Implications

This study also provides several managerial implications for managers of firms or intermediary platforms. First of all, it was shown that, in general, more solvers are preferable for the firm. Nevertheless, caused by the inverted curvilinear relationship between the number of solvers and the best contest performance, firms may want to consider to introduce a limit for contest participation. Although such an implementation contradicts the principle of open-call crowdsourcing, where everyone is invited to participate, it may result in improved contest performances. Indeed, a limitation on a first-come, first-served basis has the risk to, consequently, exclude high-expertise solvers from the contest, who would be likely to develop the best solution. Furthermore, the curvilinear relationship is determined by a very low coefficient, wherefore negative returns don't set it abrupt and neither with a strong impact. Thus, considering these two aspects, the risk of excluding skillful solvers with a registration limit and the low practical impact of the quadratic factor, an open participation should be still preferable for the firm.

As an additional contest attribute, the award structure was analysed. The results bring a few implications for contest organizers. In alignment with prior research in other types of contests (Lazear & Rosen, 1981; Snir & Hitt, 2003), the analysis indicates that a higher price award attracts more participants. Thus, firms are able to increase the level of participation by setting a higher price amount.

The impact of the monetary award on the performance in a contest reveals unexpected results. In contrast to previous studies (e.g. Archak, 2010), results show that the contest performance decreases with a higher reward. This finding is rather surprising since solvers in crowdsourcing contests are strongly motivated by the chance of winning money (Leimeister et al., 2009; Terwiesch & Xu, 2008), wherefore a higher award should raise solvers' motivations to invest efforts, resulting in better performances. This finding suggests that firms not naturally get better solutions, only by setting a high award value. One possible but rather speculative reason for this finding might be the review scorecard on the TopCoder platform, based on which the jury of a contest assesses a solver's performance (sse Appendix H). Although being very data-driven and objective, the evaluation leaves some space for subjective evaluation. When facing a contest with a high monetary price, jury members might tend to evaluate solutions slightly more negative since expectations regarding a solution may rise with a higher award.

Furthermore, findings indicate that firms can source better solutions when they design the problem description in a specific way. Firms face a conflict of interests when writing a problem description. On the one hand problem descriptions have to be written in an easy manner so that solvers from other areas of expertise are able to understand the problem (Afuah & Tucci, 2012; Spradlin, 2012). On the other hand, it is essential that problem statements include all requirements in detail so that solvers exactly understand the seeker's expectations regarding solutions. (Lüttgens et al., 2014).

The study showed that shorter problem descriptions do not only attract more solvers to a contest but also positively affect a contest's best performance. Thus, firms can achieve both, an increased level of participation and a higher contest performance by writing a concise problem description.

Finally, a high specificity of the problem description, achieved through writing a problem statement in a detailed and sophisticated way and through incorporating URL links with additional details, improve the maximum contest performance. Hence, firms should invest great efforts to provide a detailed problem description, clearly defining the problem's specifications and expectations regarding solutions of solvers.

However, in the case of the number of URL links, a trade-off was found since an increasing number of links subsequently reduces the number of participants. Therefore, firms should only insert links with additional information when they really provide valuable information regarding the problem.

6 LIMITATIONS AND FUTURE RESEARCH

The analyses of this study have a number of limitations. First of all, the sample only includes software engineering contests from two specific categories. These chosen contests may not reflect all types of crowdsourcing contests, hence the generalizability of this study may be limited. Furthermore, the focus of this study has been limited to tournament-based crowdsourcing, more specifically to expertise-based contests in which a solver's performance is mainly determined by his expertise (Terwiesch & Xu, 2008). Future research could expand the conducted analysis to other types of contests (ideation projects, trial-and-error projects) and other types of crowdsourcing methods.

Moreover, problems posted on TopCoder are rather small in terms of scale, compared with more complex, industrial problems that companies are facing. Thus, several additional factors may be relevant in regards to firms' typical problems. Nevertheless, it is worth noting that crowdsourcing contests are increasingly used by firms to solve innovation problems rather than being used sporadically for ad-hoc contests as it has historically been the case (Boudreau & Hagiu, 2009).

Another limitation is the fact that the analysis is limited to variables which are publicly available on the TopCoder website. Indeed, each contest webpage includes a forum where participants can discuss with other solvers and can get feedback on ideas from the seeking firm. This forum is not accessible for contests which took place in the past. However, the impact of such a forum on solvers' performances is expected to be negligible since an analysis of the contest sample revealed that solvers tend to submit their solutions at the very end of the contest wherefore they are not able to receive and implement feedback on time. Nevertheless, little research exists - besides a few exceptions (Füller, Hutter, Hautz, & Matzler, 2014) - that investigates the impact of collaborative features in competitive contests wherefore this is a possible subject for future research.

The expertise measurement of solvers is another possible limitation of the analysis. Solvers who have only a low expertise score may have joined TC only recently wherefore their scores don't reflect their actual skills yet. Moreover, some solvers may care more about their expertise ranking than others and are more selective in their choice of contests. Nevertheless, the Elo-based calculation of expertise is expected to be more accurate than considering absolute numbers (e.g. number of wins; number of contest participations) as another studies does (Y. Yang et al., 2010).

The analysed problems from TopCoder are software engineering problems wherefore problem descriptions have to be written in a technically sophisticated manner in order to explain the firm's problem. Thus, problem descriptions of the analysed contests are clearly at variance to other types of crowdsourcing contests, such as ideation projects (e.g. design contests). It is therefore doubtful that results can be transferred to other crowdsourcing initiatives which aim to source notably less technical solutions. Nevertheless, the major crowdsourcing platforms (TopCoder, InnoCentive, CodeChef[9]) focus on expertise-based projects, wherefore this study provides clear implications for contests on these platforms.

Future research may examine the role of the monetary award in crowdsourcing contests in more detail. Findings showed that the monetary award has a negative impact on the solvers' performances, which is contrary to prior research (Brabham, 2010; Leimeister et al., 2009). This could be explained with aspects of the motivation theory or with other factors of crowdsourcing contests, hence future research could further explore this role of the monetary award.

[9] https://www.codechef.com/

REFERENCES

Adamczyk, S., Bullinger, A. C., & Möslein, K. M. (2012). Innovation Contests: A Review, Classification and Outlook. *Creativity and Innovation Management*, *21*(4), 335–360.

Afuah, A., & Tucci, C. L. (2012). Crowdsourcing as a solution to distant search. *Academy of Management Review*, *37*(3), 355–375.

Archak, N. (2010). Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder. com. *Proceedings of the 19th International Conference on World Wide Web*, 21–30.

Bayus, B. L. (2013). Crowdsourcing New Product Ideas over Time: An Analysis of the Dell IdeaStorm Community. *Management Science*, *59*(1), 226–244.

Bockstedt, J., Druehl, C., & Mishra, A. (2015). Problem-solving effort and success in innovation contests: The role of national wealth and national culture. *Journal of Operations Management*, *36*, 187–200.

Boudreau, K. J., & Hagiu, A. (2009). Platform Rules: Multi-Sided Platforms as Regulators. *Platforms, Markets and Innoation*, 163 – 191.

Boudreau, K. J., Lacetera, N., & Lakhani, K. R. (2011). Incentives and Problem Uncertainty in Innovation Contests: An Empirical Analysis. *Management Science*, *57*(5), 843–863.

Boudreau, K. J., & Lakhani, K. R. (2013). Using the Crowd as an Innovation Partner. *Harvard Business Review*, *91*(April), 61–69.

Boudreau, K. J., Lakhani, K. R., & Lacetera, N. (2008). *Parallel Search , Incentives and Problem Type : Revisiting the Competition and Innovation Link*.

Boudreau, K. J., Lakhani, K. R., & Menietti, M. (2016). Performance Responses to Competition across Skill-levels in Rank Order Tournaments: Field Evidence and Implications for Tournament Design. *The RAND Journal of Economics*, *47*(1), 140–165.

Brabham, D. C. (2010). Moving the crowd at iStockphoto: The composition of the crowd motivation for particioation in a crowdsourcing application. *First Monday*, *13*(6), 1–19.

Breusch, T. S., & Pagan, a R. (1979). A Simple Test for Heteroscedasticity and Random

Coefficient Variation. *Econometrica*.

Bullinger, A. C., & Möslein, K. M. (2010). Innovation Contests – Where are we? *AMCIS 2010 Proceedings*, *Paper 28*, 1–9.

Casas-Arce, P., & Martinez-Jerez, F. A. (2009). Relative Performance Compensation, Contests, and Dynamic Incentives. *Management Science*, *55*(8), 1306–1320.

Cason, T., Masters, W., & Sheremeta, R. (2010). Entry into winnter-take-all and proportional-prize contests: An experimental study. *Journal of Public Economics*, *94*(9), 604–611.

Che, Y.-K., & Gale, I. (2003). Che Gale 2003.pdf. *The American Economic Review*, *93*(3), 646–671.

Chesbrough, H. W. (2003). *Open Innovation. The New Imperative for Creating and Profiting from Technology* (Vol. 1).

Chesbrough, H. W. (2011). The Era of Open Innovation. *Sloan Management Review*, 35–41.

Cohen, C., Kaplan, T. R., & Sela, A. (2008). Optimal rewards in contests. *RAND Journal of Economics*, *39*(2), 434–451.

Cohen, W. M. (1989). Innovation and learning: the two faces of R&D. *The Economic Journal*, *99*(397), 569–596.

Dahan, E., & Mendelson, H. (2001). Extreme-Value Model Concept Testing of. *Management Science*, *47*(1), 102–116.

Diener, K., & Piller, F. T. (2013). The Market for Open Innovation. *Lulu Publishing, Raleigh, NC, USA.*, *80*(10), 80–+.

Franke, N., & Piller, F. (2004). Value creation by toolkits for user innovation and design: The case of the watch market. *Journal of Product Innovation Management*.

Füller, J., Hutter, K., Hautz, J., & Matzler, K. (2014). User Roles and Contributions in Innovation-Contest Communities. *Journal of Management Information Systems*, *31*(1), 273–308.

Fullerton, R. L., & McAfee, R. P. (1999). Auctionin Entry into Tournaments. *Journal of Political Economy*, *107*(3), 573–605.

Garcia, S. M., & Tor, A. (2009). The N-Effect. *Psychological Science*, *20*(7), 871–877.

Girotra, K., Terwiesch, C., & Ulrich, K. T. (2010). Idea Generation and the Quality of the Best Idea. *Management Science*, *56*(4), 591–605.

Greenberg, M. D., Pardo, B., Hariharan, K., & Gerber, E. M. (2013). Crowdfunding support tools: predicting success & failure. In *CHI EA '13 CHI '13 Extended Abstracts on Human Factors in Computing Systems* (pp. 1814–1820).

Haller, J. B. A., Bullinger, A. C., & Möslein, K. M. (2011). Innovation contests: An IT-based tool for innovation management. *Business and Information Systems Engineering*, *3*(2), 103–106.

Hlavac, M. (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package. Retrieved from http://cran.r-project.org/package=stargazer

Howe, J. (2006). The Rise of Crowdsourcing. Retrieved April 14, 2016, from http://www.wired.com/2006/06/crowds/

Howe, J. (2008). CROWDSOURCING Why the Power of the Crowd is Driving the Future of Business. *Achievement THE International INSTITUTE*.

Jeppesen, L. B., & Lakhani, K. R. (2010). Marginality and Problem-Solving Effectiveness in Broadcast Search. *Organization Science*, *21*(5), 1016–1033.

Klemperer, P. (1995). Competition When Consumers Have Switching Costs: An Overview with Applications to Industrial Organization, Macroeconomics, and International Trade. *Review of Economic Studies*, *62*(4), 515–539.

Lakhani, K. R., & Jeppesen, L. B. (2007). Getting unusual suspects to solve R&D puzzles. *Harvard Business Review*, *85*(5), 30–32.

Lakhani, K. R., Lohse, P. a, Panetta, J. a, & Jeppesen, L. B. (2007). The Value of Openness in Scientific Problem Solving. *Biotech Business*, *18S*(Spec No 2), 5533–464.

Lampel, J., Jha, P. P., & Bhalla, A. (2012). Test-Driving the Future: How Design Competitions Are Changing Innovation. *Academy of Management Perspectives*, *26*, 71–85.

Laursen, K., & Salter, A. (2006). Manufacturing Firms THE ROLE OF OPENNESS OPEN FOR INNOVATION : IN EXPLAINING INNOVATION PERFORMANCE AMONG U . K . MANUFACTURING FIRMS. *Strategic Management Journal*, *27*(2), 131–150.

Lazear, E. P., & Rosen, S. (1981). Rank-Order Tournaments as Optimum Labor Contracts. *Journal of Political Economy*, *89*(5), 841.

Leimeister, J. M., Huber, M., Bretschneider, U., & Krcmar, H. (2009). Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition. *Journal of Management Information Systems*, *26*(1), 197–224.

Lüttgens, D., Pollok, P., Antons, D., & Piller, F. (2014). Wisdom of the Crowd and Capabilities of a Few: Internal Success Factors of Crowdsourcing for Innovation. *Journal of Business Economics*, *84*(3), 339–374.

Malone, T. W., Laubacher, R. J., & Johns, T. (2011). THE AGE OF HYPER SPECIALIZATION. *Harvard Business Review*, *89*(7/8), 56–65.

Mention, A. L. (2011). Co-operation and co-opetition as open innovation practices in the service sector: Which influence on innovation novelty? *Technovation*, *31*(1), 44–53.

Nelson, R. R. (1961). Uncertainty, Learning, and the Economics of Parallel Research and Development Efforts. *The Review of Economics and Statistics*, *43*(4), 351–364.

Piller, F., & West, J. (2014). Firms, Users, and Innovation. In *New Frontiers in Open Innovation* (pp. 29–49).

Salter, A., Ter Wal, A. L. J., Criscuolo, P., & Alexy, O. (2014). Open for Ideation: Individual-Level Openness and Idea Generation in R and D. *Journal of Product Innovation Management*.

Shao, B., Shi, L., Xu, B., & Liu, L. (2012). Factors affecting participation of solvers in crowdsourcing: An empirical study from China. *Electronic Markets*, *22*(2), 73–82.

Smith, E. a, & Senter, R. J. (1967). Automated readability index. *AMRL-TR. Aerospace Medical Research Laboratories (6570th)*, 1–14.

Snir, E. M., & Hitt, L. M. (2003). Costly Bidding in Online Markets for IT Services. *Management Science*, *49*(11), 1504–1520.

Spradlin, D. (2012). Are you solving the right problem? *Harvard Business Review*, *90*(9).

Taylor, C. R. (1995). Digging for Golden Carrots: An Analysis of Research Tournaments. *The American Economic Review*, *85*(4), 872–890.

Terwiesch, C., & Ulrich, K. (2009). *Creating, Selecting, and Developing Exceptional Opportunities*. Harvard Business Review Press.

Terwiesch, C., & Xu, Y. (2008). Innovation contests, open innovation, and multiagent problem solving. *Management Science*, *54*(9), 1529–1543.

Topcoder. (n.d.-a). Community Overview. Retrieved May 5, 2016, from https://www.topcoder.com/community/members/

Topcoder. (n.d.-b). Topcoder Review Process. Retrieved May 5, 2016, from https://www.topcoder.com/member-onboarding/topcoders-peer-review-process/

Topcoder. (n.d.-c). Topcoder's Rating System. Retrieved May 5, 2016, from https://www.topcoder.com/member-onboarding/understanding-your-topcoder-rating/

Van Der Maas, H. L. J., & Wagenmakers, E. J. (2005). A psychometric analysis of chess expertise. *American Journal of Psychology*.

von Hippel, E. (1986). Lead users: A source of novel product concepts. *Management Science*, *32*(7), 791–805.

von Hippel, E., & von Krogh, G. (2013). Crossroads - Identifying Viable "Need-Solution Pairs": Problem Solving without Problem Formulation. *Von Hippel, Eric von Krogh, Georg*, (January 2016), 1–28.

West, J., & Bogers, M. (2014). Leveraging External Sources of Innovation : A Review of Research on Open Innovation *. *Journal of Product Innovation Management*, *31*(4), 814–831.

West, J., Salter, A., Vanhaverbeke, W., & Chesbrough, H. (2014). Open innovation: The next decade. *Research Policy*.

Xu, A., Yang, X., Rao, H., Fu, W., Huang, S., & Bailey, B. P. (2014). Show Me the Money ! An Analysis of Project Updates during Crowdfunding Campaigns. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (pp. 591–600).

Yang, J., Adamic, L., & Ackerman, M. (2008). Crowdsourcing and knowledge sharing: strategic user behavior on taskcn. In *Proceedings of the 9th ACM conference on Electronic commerce* (pp. 246–255).

Yang, Y., Chen, P., & Banker, R. (2010). Impact of Past Performance and Strategic Bidding on Winner Determination of Open Innovation Contest. *Workshop on Information Systems and Economics*, 1–29.

Yang, Y., Chen, P., & Pavlou, P. (2009). Open Innovation : An Empirical Study of Online Contests. *ICIS 2009 Proceedings*, 16.

Zheng, H., Li, D., & Hou, W. (2011). Task Design, Motivation, and Participation in Crowdsourcing Contests. *International Journal of Electronic Commerce*, *15*(4), 57–88.

Ziss, S. (1994). Strategic R & D with Spillovers, Collusion and Welfare. *The Journal of Industrial Economics*, *42*(4), 375–393.

APPENDIX

Appendix A: Descriptive Statistics

Table 4: Descriptive Statistics of the Variables

Variable	N	Mean	St. Dev.	Min	Max
Contest Performance					
Maximum Score (MS)	913	92.62	5.95	73.89	100.00
Average Score (AS)	913	88.65	8.04	43.84	100.00
Contest Characteristics					
Number of Registered Solvers (NRS)	913	23.46	10.10	1.00	60.00
Contest Duration (CD)	913	5.42	1.67	0.50	17.00
Monetary Award (MA)	913	1,958.27	646.49	300.00	5,250.00
Skill Variety (SV)	913	4.75	1.80	0.00	13.00
Difficulty (D)	913	0.12	0.10	0.02	1.00
Solvers' Characteristics					
Expertise Maximum (EM)	913	1,659.19	285.42	0.00	2,291.00
Expertise Average of Registered Solvers (ERS)	913	481.13	130.80	0.00	937.27
Expertise Average of Solvers with Submissions (ERSS)	913	1,128.03	443.01	0.00	2,252.00
Problem Presentation					
Description Length (DL)	913	4,956.81	3,069.70	1,116.00	25,237.00
Description Readability (DR)	913	11.37	2.15	3.23	32.98
Number of Links in the Description (NLD)	913	2.92	4.94	0.00	60.00
Platform Characteristics					
Number of Open Contests (NOC)	913	8.00	5.52	0.00	25.00
Platform Maturity (PM)	913	383.54	147.20	175.05	724.62

Appendix B: Correlation Matrix

Table 5: Correlation Matrix

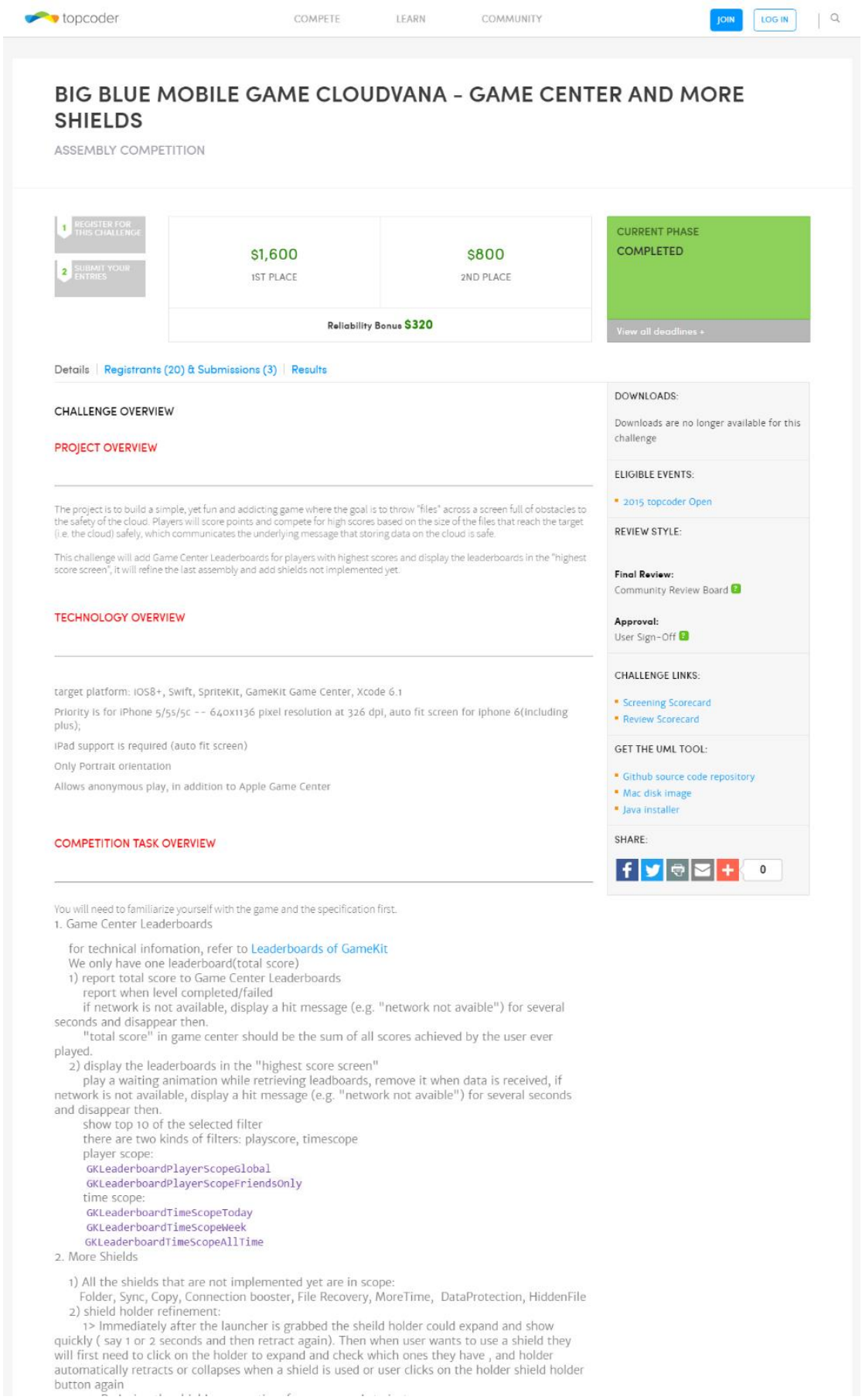
Correlation Matrix	MS	AS	NRS	NRS squared	CD	MA	SV	D	EM	ERS	ERSS	DL	DR	NLD	NOC
Maximum Score (MS)															
Average Score (AS)	0.718														
Number of Registered Solvers (NRS)	-0.015	-0.192													
NRS squared	-0.012	-0.206	0.968												
Contest Duration (CD)	-0.185	-0.125	0.327	0.288											
Monetary Award (MA)	-0.197	-0.168	0.343	0.279	0.510										
Skill Variety (SV)	-0.024	-0.049	0.132	0.129	0.002	0.011									
Difficulty (D) (log)	0.144	-0.049	-0.424	-0.326	-0.296	-0.312	-0.020								
Expertise Maximum (EM)	0.265	0.072	0.273	0.243	-0.021	0.070	0.116	-0.252							
Expertise Average of Registered Solvers (ERS)	0.188	0.205	-0.381	-0.360	-0.193	-0.178	-0.060	0.124	0.290						
Expertise Average of Solvers with Submissions (ERSS)	0.353	0.430	-0.074	-0.090	-0.093	0.040	0.106	-0.228	0.471	0.285					
Description Length (DL) (log)	-0.186	-0.292	0.099	0.112	0.155	0.268	0.158	-0.016	0.189	-0.102	0.037				
Description Readability (DR) (log)	0.053	0.078	0.009	0.010	0.064	0.081	0.010	-0.065	0.003	0.016	0.031	0.052			
Number of Links in the Description (NLD)	0.087	0.138	-0.261	-0.220	-0.113	-0.174	0.044	0.143	-0.022	0.224	0.003	0.110	-0.064		
Number of Open Contests (NOC)	0.146	0.273	-0.364	-0.360	-0.015	-0.051	-0.195	0.037	-0.093	0.291	-0.030	-0.241	0.089	0.096	
Platform Maturity (PM) (log)	-0.040	-0.089	0.574	0.531	0.111	0.163	0.151	-0.277	0.087	-0.450	-0.001	0.149	-0.081	-0.147	-0.503

Appendix C: Variance Inflation Factors

Table 6: Variance Inflation Factors of independent Variables

	Dependent variable:			
	Number of Registered Solvers (NRS)	Average Score (AS)	Maximum Score (MS)	
	1	2	3.1	3.2
Number of Registered Solvers (NRS)		2.426	2.426	24.728
NRS (squared)				19.437
Contest Duration (CD)	1.358	1.499	1.499	1.499
Monetary Award (MA)	1.447	1.534	1.534	1.589
Skill Variety (SV)	1.078	1.089	1.089	1.089
Difficulty (D) (log)		1.753	1.753	1.845
Expertise Maximum (EM)		1.823	1.823	1.878
Expertise Average of Registered Solvers (ERS)		1.731	1.731	1.731
Expertise Average of Solvers with Submissions (ERSS)		1.701	1.701	1.723
Description Length (DL) (log)	1.204	1.273	1.273	1.295
Description Readability (DR) (log)	1.032	1.033	1.033	1.034
Number of Links in the Description (NLD)	1.116	1.172	1.172	1.184
Platform Maturity (PM) (log)	1.372	1.810	1.810	1.863
Number of Open Contests (NOC)	1.412			

Appendix D: TopCoder Contest



topcoder COMPETE LEARN COMMUNITY JOIN LOG IN

BIG BLUE MOBILE GAME CLOUDVANA - GAME CENTER AND MORE SHIELDS

ASSEMBLY COMPETITION

1 REGISTER FOR THIS CHALLENGE
2 SUBMIT YOUR ENTRIES

$1,600 1ST PLACE
$800 2ND PLACE
Reliability Bonus $320

CURRENT PHASE
COMPLETED
View all deadlines +

Details | Registrants (20) & Submissions (3) | Results

CHALLENGE OVERVIEW

PROJECT OVERVIEW

The project is to build a simple, yet fun and addicting game where the goal is to throw "files" across a screen full of obstacles to the safety of the cloud. Players will score points and compete for high scores based on the size of the files that reach the target (i.e. the cloud) safely, which communicates the underlying message that storing data on the cloud is safe.

This challenge will add Game Center Leaderboards for players with highest scores and display the leaderboards in the "highest score screen", it will refine the last assembly and add shields not implemented yet.

TECHNOLOGY OVERVIEW

target platform: iOS8+, Swift, SpriteKit, GameKit Game Center, Xcode 6.1

Priority is for iPhone 5/5s/5c -- 640x1136 pixel resolution at 326 dpi, auto fit screen for iphone 6(including plus);

iPad support is required (auto fit screen)

Only Portrait orientation

Allows anonymous play, in addition to Apple Game Center

COMPETITION TASK OVERVIEW

You will need to familiarize yourself with the game and the specification first.

1. Game Center Leaderboards

for technical infomation, refer to Leaderboards of GameKit
We only have one leaderboard(total score)
1) report total score to Game Center Leaderboards
report when level completed/failed
if network is not available, display a hit message (e.g. "network not avaible") for several seconds and disappear then.
"total score" in game center should be the sum of all scores achieved by the user ever played.
2) display the leaderboards in the "highest score screen"
play a waiting animation while retrieving leadboards, remove it when data is received, if network is not available, display a hit message (e.g. "network not avaible") for several seconds and disappear then.
show top 10 of the selected filter
there are two kinds of filters: playscore, timescope
player scope:
`GKLeaderboardPlayerScopeGlobal`
`GKLeaderboardPlayerScopeFriendsOnly`
time scope:
`GKLeaderboardTimeScopeToday`
`GKLeaderboardTimeScopeWeek`
`GKLeaderboardTimeScopeAllTime`

2. More Shields

1) All the shields that are not implemented yet are in scope:
Folder, Sync, Copy, Connection booster, File Recovery, MoreTime, DataProtection, HiddenFile
2) shield holder refinement:
1> Immediately after the launcher is grabbed the sheild holder could expand and show quickly (say 1 or 2 seconds and then retract again). Then when user wants to use a shield they will first need to click on the holder to expand and check which ones they have , and holder automatically retracts or collapses when a shield is used or user clicks on the holder shield holder button again

DOWNLOADS:
Downloads are no longer available for this challenge

ELIGIBLE EVENTS:
- 2015 topcoder Open

REVIEW STYLE:
Final Review:
Community Review Board
Approval:
User Sign-Off

CHALLENGE LINKS:
- Screening Scorecard
- Review Scorecard

GET THE UML TOOL:
- Github source code repository
- Mac disk image
- Java installer

SHARE: 0

2> Reducing the shields pop up time from 5 seconds to just 3
3. Update marketing/commercial messages(displayed in loading screen)
when the app starts up, it will send an http request(with local version number) to get latest version and all messages from remote server(if the version is already the latest, the remote server will simply response with no messages) and store in Messages.plist, if failed(e.g. network error), still use the old messages.plist(update the default messages.plist with the 1st and 2nd messages in cloud facts.txt)
In loading screen, one of the N messages will be chosen randomly to display.
The remote server could be a NodeJS server provides version check and message feedback services.
4. Game scene refinement
use the score text at the top of the game screen as a sort of progress bar (for instance with green color so when you achieve the goal for a given level, the entire word "score" is turned "green")
5. Obstacle refinement
1) Enhance the obstacle "Dark clouds & lightning", bolts coming from the dark clouds should be random directions (right now they all seem to come left oriented)
2) Support more fields for obstacle in level configure:
f1 - id: required, int, id for reference.
f2 - notLoadOnStart: optional, bool, if true, the obstacle won't appear after level is loaded.
f3 - tdelay: optional, int, time to delay, if exists, the obstacle won't appear after level is loaded, instead, it will appear after "tdelay" seconds if the level is still active.
f4 - ttl: optional, int, time to live, if exists, the obstacle will be destroyed after "ttl" seconds (and ttlToBe is missing).
f5 - ttlToBe: optional, array of int, id of obstacle to appear after ttl seconds, if exists, when the obstacle is destroyed after "ttl" seconds, a new obstacle(randomly choose from ttlToBe) is created at the same position, make sure the transition is smooth
the client need to try one level maybe on the higher ones that is truly random and has obstacles showing and vanishing so they change during the same level, configure the level to make all the obstacles have a chance to appear.
6. Bug fix
when exiting game while on pause and going back in again , pause screen is displayed but timer is not stopping (obstacles animations are also resumed). Steps to reproduce:
1) While playing any level click on pause, Pause screen and vertical menu bar pops up all action and timer is frozen as expected.
2) Click on the iphone home button (exit game go back to phone home pages)
3) Then click on the PayIBM icon to go back in the game
4) Pause screen is displayed (as expected) however you will notice the timer and onstacle action in the screen is no longer paused. (should be frozen until user exits pause mode)
7. Level Selection screen
Add an entry in global configure: dev
if it is true, then in level selection screen
1) if touch on a unlocked level cloud for more than e.g. 5 seconds, all the open levels are locked except the first level.
2) if touch on a locked level cloud for more than e.g. 5 seconds, all the locked levels(level <= the touched level) are unlocked(with zero star)
8 Level Configures
Add an entry in global configure: levelset
It is a dir name where to load level configuration files.
e.g. we may have following sets:
dev: add level configure for each newly added obstacle of this challenge here (include the level described in 5.2: Support more fields for obstacle in level configure) , increase goal specially on higher levels.
product: product levels coming later.

GENERAL NOTICE

---

1. Memory leaks check: There should be no memory leaks for this app. Please use Instrument to check memory leaks, espcialy reset game repeatly
2. We use texture packer (*https://www.codeandweb.com/**texturepacker***) to packer sprite images (except backgrand images), please check atlas_projects of last assembly submssion.

PLATFORMS

iOS

TECHNOLOGIES

Swift Xcode iOS

Figure 7: Exemplary TopCoder Contest Description

Appendix E: TopCoder Participants

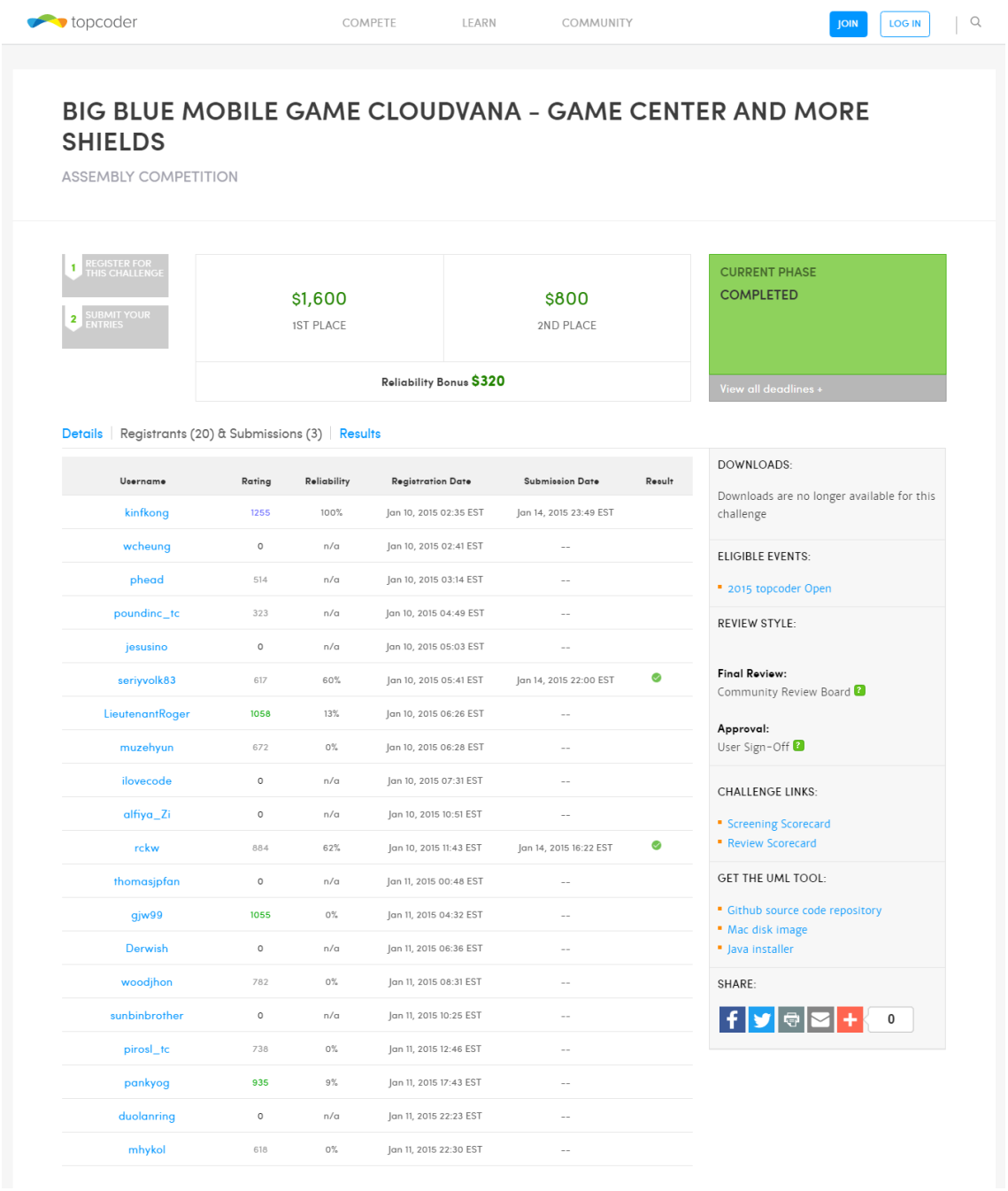


Figure 8: List of participating Users in a TopCoder Contest

Appendix F: TopCoder Results

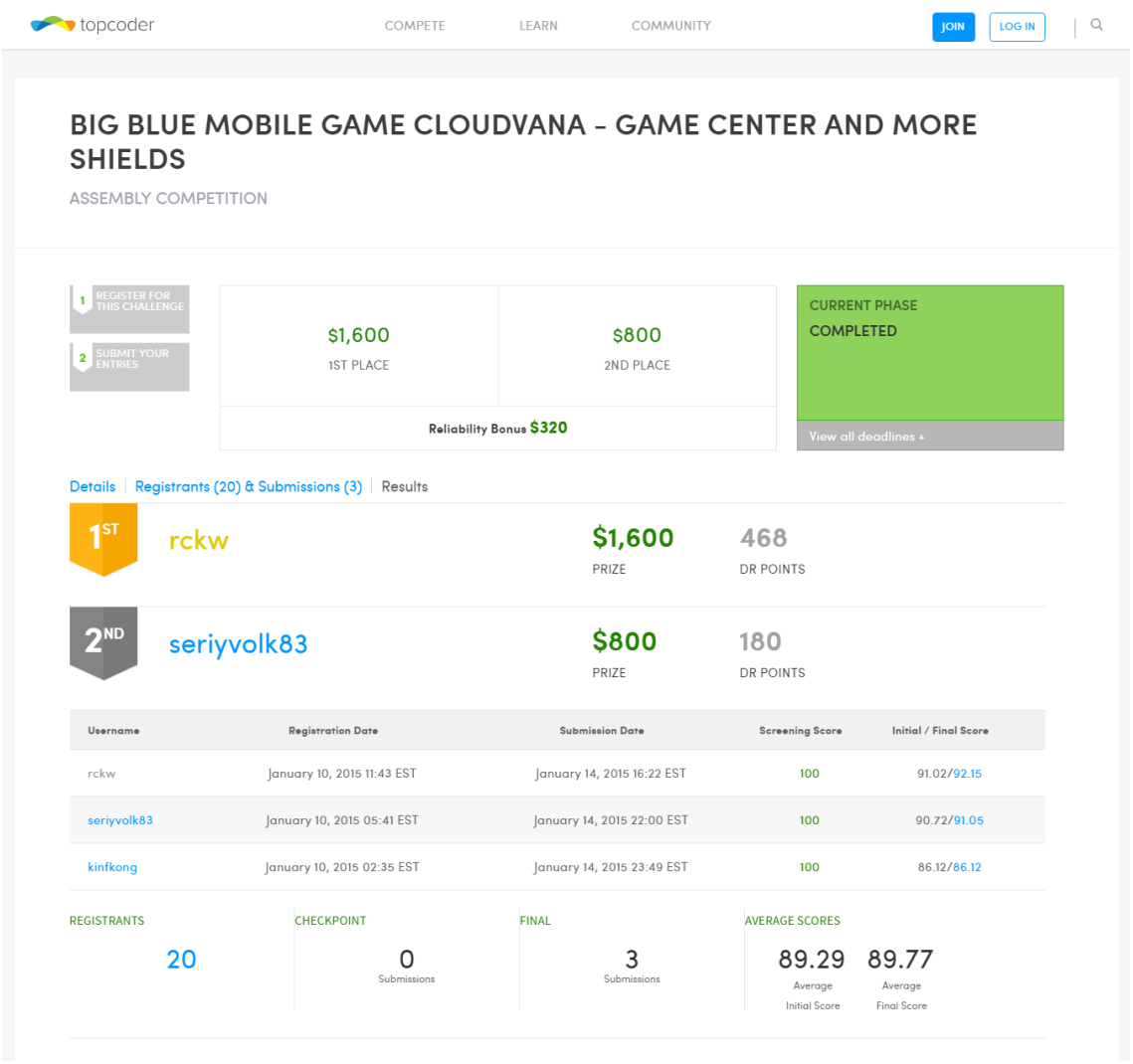


Figure 9: TopCoder Results Overview

Appendix G: TopCoder Solver Profile

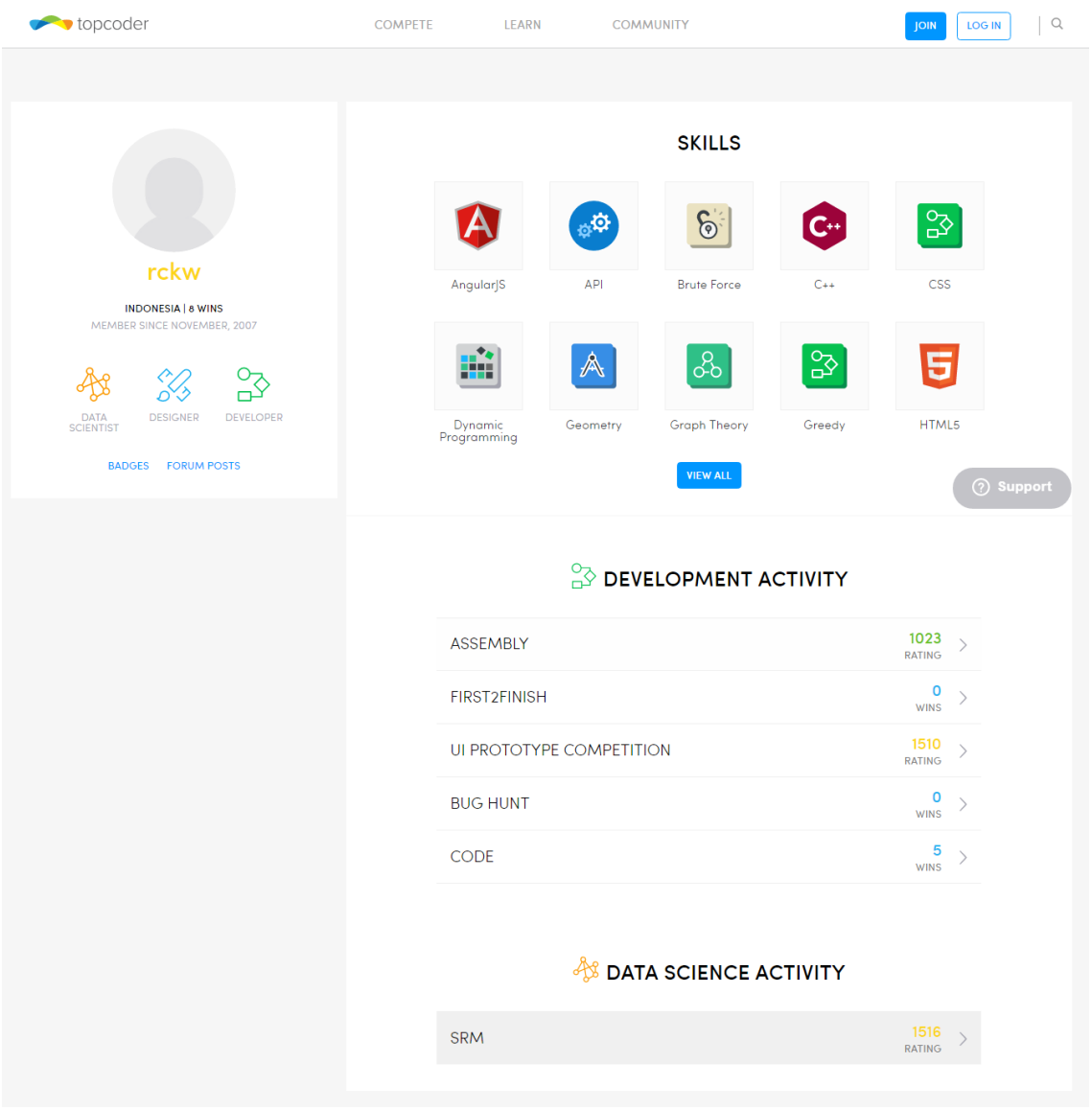


Figure 10: TopCoder Solver Profile

Appendix H: TopCoder Review Scorecard

Module Assembly Review Scorecard

The minimum passing score: 80.0

General Assembly (65.0)		
Functional Requirements (70.0)	**Weight**	**Response**
Question 1.1.1 Each major requirement is implemented correctly as specified in the contest specification. Major requirements are those requirements that determine the success of the application. For a GUI-driven application, all use cases where the actor is the user are considered major requirements. For a system-driven application, all background process use cases are considered major requirements. Major requirements are the most important requirements of any application. For example, in an application that allows users to "upload", "download", "review" and "watch" final-fix submissions, the "upload submission", "download submission" and "review submission" use cases could be considered major requirements while the "watch final-fix submission" use case, which sends an email when a new version of the submission being watched is uploaded, could be considered minor since it is not critical to the success of the application. The "upload" use case is an especially significant major requirement since the "download" and "review" use cases depend on it; submissions to be downloaded for review must first exist in the application, which can only happen if all "upload" use cases complete successfully. Unless the contest spec states otherwise, minor requirements will generally include: configuration, logging, auditing, concurrency, transaction management and exception handling, which are all standard application concerns. Such standard application requirements, unless explicitly listed as major in the contest spec, are not evaluated in this question. Additionally, an implementation may have addressed several minor requirements, such as the "watch final-fix submission" use case in the example above, but if it did not address the major requirements then it must be given a score of 1 in this question. If it is not easily apparent from the contest input what constitutes major requirements, reviewers must first seek confirmation in the forum or via "Contact Managers" before making their judgment. Rating 1 - Multiple major requirements are missing or were improperly implemented, and most major functionality is missing or does not work. Rating 2 - A few major requirements are missing or were improperly implemented. Significant areas of the major functionality are missing or do not work. Rating 3 - The implementation takes into consideration all major requirements, but there are some areas where one or more major requirements are not fully addressed and necessary features are missing. Rating 4 - The implementation fully addresses all major requirements. All major functionality can be used without any issues.	80.0	
Question 1.1.2 Each minor requirement was implemented correctly as specified in the contest specification. Please refer to the previous question for guidance on how to determine when a requirement can be regarded as minor. Standard application requirements are to be evaluated here. Rating 1 - Multiple minor requirements are missing or were improperly implemented, and most minor functionality is missing or does not work. Rating 2 - A few minor requirements are missing or were improperly implemented. Significant areas of the minor functionality are missing or do not work. Rating 3 - The implementation takes into consideration all minor requirements, but there are some areas where one or more minor requirements are not fully addressed and necessary features are missing. Rating 4 - The implementation fully addresses all minor requirements. All minor functionality can be used without any issues.	15.0	
Question 1.1.3 Does the application handle erroneous input gracefully? If there are specific validation requirements, from the ARS or elsewhere, they must be scored here. Rating 1 - Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	5.0	
Nonfunctional Requirements (30.0)	**Weight**	**Response**
Question 1.2.1 Can the application be deployed easily, without complications or troubleshooting, in the development environment? Is automation used effectively? If automation is not required, is there enough separation in how the build or shell scripts are written to make automated deployment easier in future? Rating 1 – Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	30.0	
Question 1.2.2 Does the implementation include all required dependencies to ensure that the application can be deployed successfully in the TARGET environment used by the client? For example, if the module will be used inside a container, it must deploy successfully in that container in addition to the possibly different environment used for development. Rating 1 - Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	25.0	
Question 1.2.3 Does the application stay usable under various edge cases? The application must stay usable under edge cases that are part of the requirements. For example if a page loads records from a backend database into a table, and the user is allowed to specify that "all" records be loaded, then the page must still respond appropriately when that option is used. Another example would be if the application is expected to support peak loads of N concurrent users. Rating 1 - The application does not support one or more edge cases and does not recover Rating 2 - The application does not support one or more edge cases, but it recovers and still functions for general use cases Rating 3 - The application usability degrades somewhat but it still responds in an appropriate fashion Rating 4 - The application stays usable and responsive under all supported edge cases	15.0	
Question 1.2.4 Is the application responsive? Do user actions complete in an appropriate amount of time? It should meet the performance requirement in concept document (or in a reasonable time if no such requirement) Reviewers should adjust these measures if it is known that the development environment itself impacts performance. Rating 1 - Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	15.0	
Question 1.2.5 Are all application data secure? If there are data entitlement constraints regarding retrieval, display and persistence of data, verify that these constraints are met in the submission. Rating 1 – Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	15.0	
Module Assembly (35.0)		

Module Assembly (35.0)		
Design Adherence (25.0)	**Weight**	**Response**
Question 2.1.1 The implementation uses all required technologies (language, framework, etc) and packages as specified in the requirement and design documents. Best practices for the technology are followed unless specified otherwise. Rating 1 - Assemblers are not familiar with the technology and it is frequently misused. Rating 2 - Technologies are implemented according to best practices, though there are occasional misuses that impact functionality or performance in some fashion. Rating 3 - Technologies are implemented according to best practices, though there are occasional misuses. Functionality or performance is not impacted. Rating 4 - Assemblers exhibit excellent knowledge and understanding of the technologies involved. Best practices are always followed.	70.0	
Question 2.1.2 The implementation properly implements configurable parameters as defined in the design documents. Ensure that the implementation uses an appropriate level of configurability to facilitate future expansion with a minimum of code change. Rating 1 - Some of the configurable parameters are hard coded. Rating 2 - The configurable parameters defined in the design documents are implemented, though some of them do not work as intended; for instance, a parameter configured in multiple locations, or a parameter that is ignored in some scenario. Rating 3 - The configurable parameters defined in the design documents are implemented exactly as intended. Rating 4 - The configurable parameters defined in the design documents are implemented exactly as intended. Additionally the assemblers made other parameters configurable where it made sense to do so and used sensible conventions and default values that minimize configuration complications.	30.0	
Code Review (25.0)	**Weight**	**Response**
Question 2.2.1 For those parts that are not covered by the design documents (architecture) the assemblers must properly adhere to basic design principles. The code must be ready to accommodate foreseeable and likely change requests. This usually refers to the utility classes, helper logic, etc. that the assemblers produce out of the architecture scope. Rating 1 - The extra code lacks design principles and has frequent redundancies. It is difficult to change. Rating 2 - The extra code contains some level of redundancy and requires considerable changes to accommodate likely foreseeable changes. Rating 3 - The assemblers understand the design principles underlying the language(s) used. The extra code contains little redundancy and accommodates changes easily. Rating 4 - The assemblers exhibit excellent design skills. The extra code written contains no code redundancy and accommodates changes easily. Appropriate design patterns are utilized.	15.0	
Question 2.2.2 The implementation chooses efficient ways to perform the functional tasks. This often requires the assemblers to understand the components and make the right decisions such as to utilize the bulk methods instead of run a method multiple times. Another example would be to use a regular expression to validate an input string rather than constructing a lengthy check block. Rating 1 - The implementation generally does not take efficiency into account. Rating 2 - The implementation is not efficient in various spots. Rating 3 - The implementation is efficient but can be improved in a few minor spots. Rating 4 - The implementation efficiently chooses the right approaches as appropriate.	15.0	
Question 2.2.3 There is no unnecessary or careless object instantiation or variable assignment. Rating 1 - There are many cases with unnecessary or careless object instantiation or variable assignment. Rating 2 - There are a number of cases with unnecessary or careless object instantiation or variable assignment. Rating 3 - There is little unnecessary or careless object instantiation or variable assignment. Rating 4 - There is no unnecessary or careless object instantiation or variable assignment at all.	5.0	
Question 2.2.4 Errors must be properly encapsulated from the end user, who should not see a straight stack trace or logging intended for the developer or the support team. Error message must be appropriate so that user knows how to react, and can properly engage administrators or a support team to reproduce and diagnose the problem. Rating 1 - End user observes stack traces frequently or sees error messages that are entirely unhelpful. Rating 2 - Most errors are encapsulated from the end user, but the error messages are either too general or too low level. Rating 3 - Exceptions are fully encapsulated from the end user. Sometimes error messages are not accurate or shared by errors with different causes. Rating 4 - Exceptions are fully encapsulated from the end user. Error messages are appropriate.	10.0	
Question 2.2.5 Application must produce the proper logs for the administrators to utilize. Reviewer should not only examine whether appropriate logging messages are produced, but should also check if the right logging levels are used. - Logs should appropriately record application state and supporting values to help diagnose issues. - Logs should not contain irrelevant information. For example instance names that are JVM generated. - Logs should make appropriate use of standard logging levels. - Debug logging should include complex decision points. Rating 1 - Application hardly produces any useful logging. Rating 2 - Application lacks necessary logging in some places. Logging levels are not used correctly. Rating 3 - Application produces detailed and useful logging but the logging levels are not usually appropriate. Rating 4 - Application produces precise and useful logging.	10.0	
Question 2.2.6 Application must adhere to the proper transaction level as specified by the design documents. Rating 1 - The implementation does not take transactions into account. Rating 2 - The transaction implementation adheres to the design documents. There are some use cases where transactions are ignored or implemented incorrectly. Rating 3 - The transaction implementation adheres to the design documents. There are a few use cases that are not fully transactional. Rating 4 - The transaction implementation adheres to the design documents. Every transaction is properly handled.	10.0	
Question 2.2.7 Application must protect against malicious or potentially unsafe code. The code cannot be injected with malicious input that can easily be exploited by common injection attacks. The code does not use potentially unsafe constructs as part of core application logic. For instance, commands like "rm -rf /" or "DROP DATABASE" if used carelessly, can cause data loss. Rating 1 - Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	5.0	
Question 2.2.8 There is no duplication of code observed in the implementation, including code written for automated tests. Code is appropriately refactored to eliminate redundancy. Rating 1 - Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	5.0	
Question 2.2.9 The implementation code contains detailed and useful documentation. Rating 1 - The implementation code contains little documentation to the extent that it's difficult to understand the code. Rating 2 - The implementation code contains documentation for publicly accessible interfaces such as public classes and methods, but the documentation is not helpful. For example, on a public method simply restating the names of the input parameters and their types is useless. Rating 3 - The implementation code contains documentation for publicly accessible interfaces such as public classes and methods. The documentation clearly explains how to use and what to expect from the publicly accessible interface points. Internal logic such as private methods, or complex pieces of code behind a public interface are left unexplained such that someone who has to modify (as opposed to consume) the code will have difficulty. Rating 4 - The implementation code contains documentation for publicly accessible interfaces such as public classes and methods. The documentation clearly explains how to use and what to expect from the publicly accessible interface points. Additionally internal logic is well documented such that future developers can understand what the code is doing.	20.0	
Question 2.2.10 All code, including test cases, follows the TopCoder coding conventions as documented in the General Assembly Tutorial. http://apps.topcoder.com/wiki/display/tc/General+Assembly+Competition+Tutorial#GeneralAssemblyCompetitionTutorial-Toc358837432. The reviewer must check the code style used by the developer. There must be no tabs, proper tags (author, version) and copyright section. Rating 1 - Consistent code violations are present to such a degree that the code looks bad. Rating 2 - Many code violations are present, but when looking at the code it still is professional. Rating 3 - Some minor code violations are present here and there. Rating 4 - The code style is perfect and every element is present.	5.0	

Automated Tests (15.0)	Weight	Response
Question 2.3.1 Automated tests such as unit test cases thoroughly test all methods and constructors. Rating 1 - Some of the core methods or constructors are not tested. Rating 2 - Most of the methods and constructors are tested, with some trivial ones left untested. Rating 3 - All methods and constructors are tested. Some of the constructors or methods share the same test case. Rating 4 - All methods and constructors correspond to at least one test case.	25.0	
Question 2.3.2 Test cases effectively test the implementation. This usually requires multiple test cases for the more complicated methods to cover each logic path separately. Test data must be carefully chosen to enumerate the possibilities. Rating 1 - Test cases are stylistic and have little value in testing the real functionality. Rating 2 - Test cases are generally effective though some of the complicated cases are tested superficially. Rating 3 - Test cases are well written to effectively test the implementation. Some unimportant execution paths are not covered. Rating 4 - Test cases are well written to effectively test the implementation. Each complicated methods are tested with multiple sets of data as necessary.	40.0	
Question 2.3.3 Where applicable, unit test cases properly make use of setup and teardown methods to configure the test environment. Rating 1 - The test cases require manual setup of the system for each run. Rating 2 - The test cases occasionally leaves the environment unrestored (leaving temporary data or file, etc.), but test cases can still be run again. Rating 3 - The environment is properly configured by setup and teardown methods. The environment is fully restored after the test case is executed. Rating 4 - The environment is properly configured by setup and teardown methods. Failure of a single test case does not affect any of the other test cases.	10.0	
Question 2.3.4 The test code contains documentation for classes, methods and variables. Rating 1 - The test code is not documented. Rating 2 - The test code contains some documentation for classes, methods and variables. Rating 3 - The test code contains majority of the documentation for classes, methods and variables. Rating 4 - The test code contains documentation for classes, methods and variables.	25.0	
Test Data (10.0)	**Weight**	**Response**
Question 2.4.1 The submission includes useful test data to verify the implementation works as designed. For complex data models, ensure that the included tests are non-trivial otherwise the submission must include guidance on how to perform non-trivial testing. If the test data was not required, the maximum score must be given here. Rating 1 – Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	100.0	
Deployment Guide (25.0)	**Weight**	**Response**
Question 2.5.1 Deployment guide is well written and details the steps to configure and deploy the application. Rating 1 - Deployment guide is missing or lacks core information to configure and deploy the application. Rating 2 - Deployment guide describes the steps to configure and deploy the application, though the steps are not complete and thus require some amount of external research. Rating 3 - Deployment guide describes the steps to configure and deploy the application, however missed some points that require some minor guess or external research. Rating 4 - Deployment guide is precise and fully describes the steps to configure and deploy the application.	45.0	
Question 2.5.2 The deployment guide contains easy to follow verification steps to properly test the implementation. In steps where screen shots are used, reviewers must verify the ability to reproduce the information shown and that the actual output is accurate. No personally identifiable information may be revealed unless it is necessary to verify a feature is working as designed. Rating 1 - Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	35.0	
Question 2.5.3 All assets are named clearly and appropriately. Grammatical errors, spelling mistakes, formatting and copy-paste mistakes are to be scored here. Also pay attention to typos in file names. Rating 1 – Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	5.0	
Question 2.5.4 The submission uses only allowable file formats listed in the Assembly Tutorial. Proper justification must be provided if there are significant deviations from the recommended formats. Rating 1 – Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	5.0	
Question 2.5.5 The submission is organized in the format recommended by the Assembly Tutorial. Proper justification must be provided if there are significant deviations from the recommended formats. Rating 1 – Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	5.0	
Question 2.5.6 The deployment guide does not contain inaccurate or redundant information. Verify that product versions used match the versions indicated in the contest specification. Rating 1 – Disagree Rating 2 - Somewhat Agree Rating 3 - Agree Rating 4 - Strongly Agree	5.0	

Figure 11: TopCoder Review Scorecard for Assembly Contests

Appendix I: Crawling

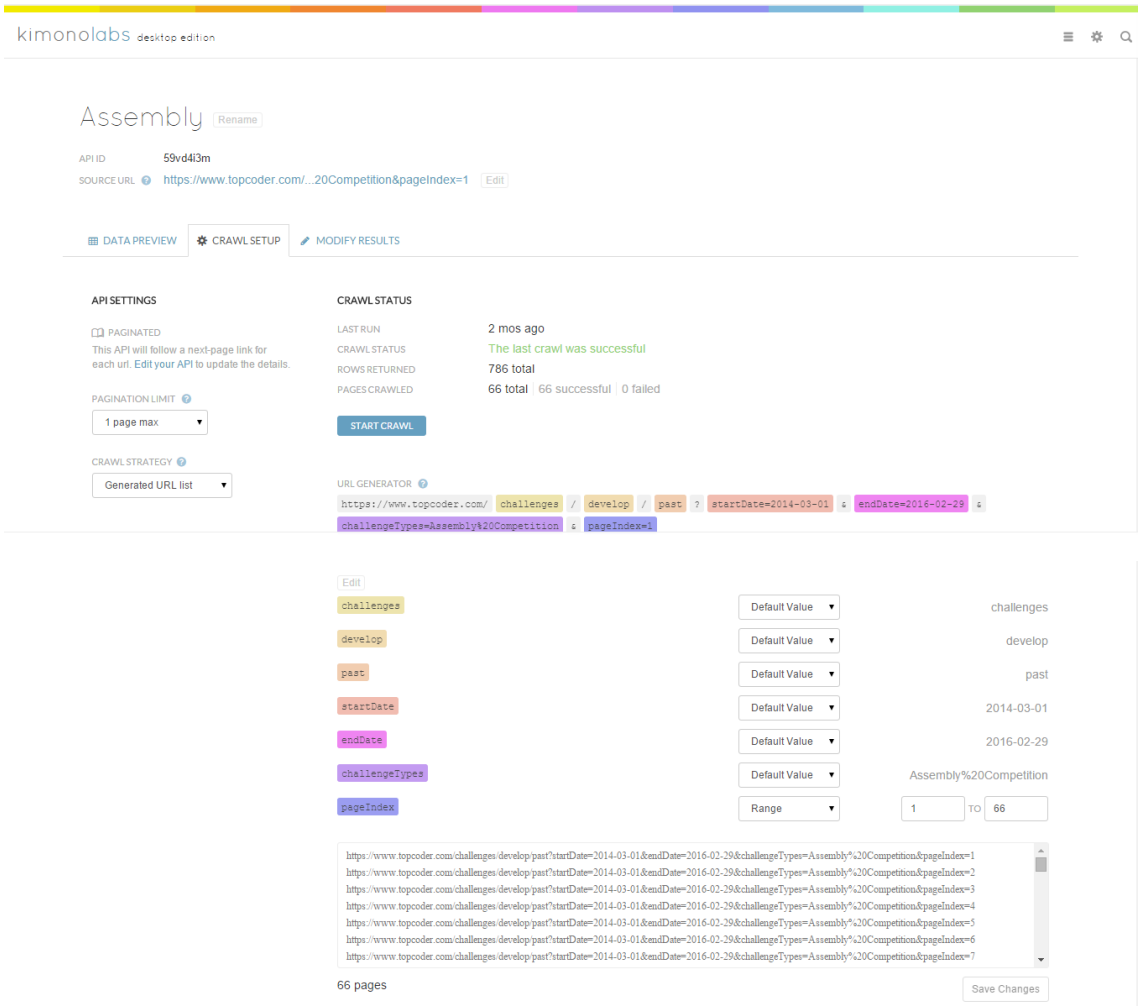


Figure 12: Exemplary Crawling Setup for Assembly Contests