IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 17, 2019, accepted May 3, 2019, date of publication May 8, 2019, date of current version May 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915604

An Empirical Study on Competitive Crowdsource Software Development: Motivating and Inhibiting Factors

INAM ILLAHI[1], HUI LIU[1], QASIM UMER[1], AND SYED ANEES HAIDER ZAIDI[2]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2]Department of Management Science and Engineering, School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Hui Liu (liuhui08@bit.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61690205 and Grant 61772071.

ABSTRACT Crowdsourcing is gaining more and more popularity among the academic and industrial community. Organizations are adopting this technological advent and increasingly crowdsourcing their tasks to the unknown individuals. However, in the context of competitive crowdsourcing software development (CCSD), crowdsourcing is still unexplored. Too little is presently known about what intricate developers to participate in crowdsourcing software development competitions. Most importantly, what kind of developers are more likely to participate? Such open questions remain to be explored. To this end, in this paper, we present the results of an empirical study conducted to investigate what motivates software developers to participate in CCSD and what inhibits software developers to participate in such competitions. An online questionnaire is sent out to more than 300 crowdsource software participants, of which 113 return valid responses. It is also sent to more than 150 industry practitioners, of which 75 return valid responses. The results suggest that the monetary rewards are not significantly important to motivate software developers to participate in CCSD. Instead, learning, social contacts, and peer recognition are more important. Besides the survey, we also analyze the historical data collected from one of the most popular software crowdsourcing platforms. The analysis results reveal that the Pareto principle holds for CCSD as well, and 0.9% of the participants win 86% competitions. The results support the premise that CCSD market is still at an early stage. Most of the professional software engineers do not participate seriously in crowdsourcing software development. Therefore, many crowdsourced tasks, especially complex tasks, may fail to receive any satisfying submission. These findings are worthwhile for the crowdsourcing platforms and companies who want to outsource their software development tasks to the CCSD platforms.

INDEX TERMS Crowdsourcing, motivation, inhibiting factors, competitive software development.

I. INTRODUCTION

Crowdsourcing has earned a reputation as an effective and quick way of software development [1], [2]. Initially, crowdsourcing was started in 1991 when a student, Linus Torvalds, from University of Helsinki, Finland invited software developers around the world to participate in an open-source operating system. Linux 1.0 was the first crowdsourcing software project that published in 1994. Later in 2006 Howe and Robinson coined the term '*crowdsourcing*' [3].

Competitive crowdsource software development (CCSD) is an emerging trend to outsource software development tasks and has gained significant attention of the software engineering community [3]. The ecosystem of CCSD mainly depends on the unknown, geographically distributed and uncontrolled crowds. In CCSD, a project is divided into several competitive tasks [4] and posted on a platform. Through an open call globally distributed software crowd compete to provide the best solution for the given task and only one or two winners are rewarded and others get no monetary rewards. Many crowdsourcing platforms such as TopCoder [5], CodeForces [6], HackerEarth [7], CodeChef [8], Hackerrank [9], SPOJ [10], Atcoder [11], Kaggle [12], and InnoCentive [13] frequently announce software development competitions pertaining to complex tasks and seeking innovative solutions. However, among all TopCoder is the most popular platform with the global community of 1.2 million registered members.

The associate editor coordinating the review of this manuscript and approving it for publication was Yongwang Zhao.

Some previous studies spectacle the fast growth of these online working websites [14] and the popularity of the crowd contests [15]. However, empirical evidence [16] shows a gap between the market demand and the supply of competitive crowd workers for software development. Many software development tasks do not get satisfying submissions [17]. Consequently, requesters received low-quality [18] and less innovative software solutions [19] from crowdsourcing platforms such as TopCoder. For instance, on TopCoder the submission failed review rate is (35.4%) [17], conclusively there are around (28%) software development challenges that either get cancelation due to no registrants or do not get any satisfying solution. This indicates the potential failure factors associated with motivating factors e.g., monetary rewards [20], [21].

The motivation of the software crowd is critical to the success of CCSD. It determines the quality and the quantity of the contributions of the crowd [22], [23] and considered to be a major factor in project success [24], [25]. In order to motivate the crowd to work on the offered tasks, some financial or social incentives are rewarded to them [26]. The subject of crowds' motivation has been widely studied. Several researchers [14], [23], [27]–[31] have explored crowdsourcing focusing on motivational models and social sciences theories.

Although plenty of research can be found on crowds' motivation, CCSD is still a subject to be explored for what motivate competitive software developers to participate in crowdsourcing software development competitions and its potential drivers and barriers. Software development tasks are often tiresome, complex, interdependent and required substantial expertise, cognitive effort and countable periods of time [18]. Existing literature lacks in empirical evidence to understand what recompense competitive software developers who participate in CCSD, where only one or two winners are rewarded with the monetary reward and others get no monetary incentive. There are some essential questions associated with CCSD remain open, e.g., how to establish motivation and trust of the crowd [23], [32], what motivates competitive software developers the most to dedicate their time, skills and knowledge to CCSD? Therefore, there is a great need to understand, what existing studies validate and what real practitioners aimed when participating in such software development competitions that may take a significant amount of time and require specific expertise to participate.

To this end, in this paper we present the results of a study conducted to investigate: 1) What motivates competitive software developers the most when participating in CCSD and 2) What inhibits software developers who don't want to participate in such competitions. 3) Does Pareto principle hold for CCSD, i.e., whether most of the contributions are made by a small part of the participants? In this perspective, an online questionnaire was served to crowdsource software development participants aiming to understand what triggers them to participate in the competition and what do they get in return when taking part in such competitions. Moreover, we analyze the history data, extracted from Topcoder, with the goal of discovering past behavior of software developers participating in software development challenges, i.e., how many software developers participated in previous software development competitions and how many submissions/contributions they have made. The main findings of the study are listed as follow:

- The Pareto principle holds for CCSD as most of the competitions are won by a small number of developers.
- Improving programming skills and learning new technologies are most potential factors for the software developers participate in CCSD.
- The monetary reward has little effect on software crowds' motivation to participate in crowdsourcing competitions.
- Lack of expertise required for competitive software development and uncertainty to win by thousands are most deviating factors inhibiting industrial software developer to participate in CCSD.

The remaining sections of the paper are organized as follows: Section II provides related work and differences, where, Section III provides explanation and articulate the differences that separate this study from the existing state of the art literature. Section IV provides the research methodology, research questions, study design, data collection, and data analysis. The results are presented in Section V. Section VI presents the implication. the threats to validity are discussed in Section VII. Section VIII includes the conclusions and future work.

II. RELATED WORK

A. CLASSIFICATION OF MOTIVATION

The subject of motivation has extensively been studied in psychology, sociology, management, and marketing [33] providing a solid theoretical ground upon human motivation [23], [29]. Researchers around the world have investigated the motivational aspect of the crowd with different theories, e.g., value theory, value expectancy theory, motivation theory, social identity theory, and value-sensitive design theory, all these have been applied to explain the participant's intention or continuance in crowdsourcing [14], [29], [32], [34]. These studies have mainly investigated the influences of benefiting factors on crowds' intention.

According to the theory developed by [35], who started working on it from the 1970s [29], motivations are of two types intrinsic and extrinsic. Motivation is intrinsic when the reward for doing an activity comes from the activity itself, for instance, playing football because it is enjoyable [27]. In the case of extrinsic motivation, the activity is just an instrument for achieving a certain, desired, outcome [27], for instance, people work hard to earn money, study for exams, or play football to win a championship [30]. However [35] refined these two categories to add up a third, called intermediate category: internalized extrinsic motivations [29] which are extrinsic motivations. In internalized extrinsic motivation the activity is assimilated by the individual to perform but

the individual has understood the benefit of his action [29]. *The more one internalizes the reasons for an action and assimilates them to the self, the more ones extrinsically motivated actions become self-determined* [35]. Recently, [30] presented 25 motivational factors identified from the literature review and categorized them into three separate categories, i.e., intrinsic motivators, extrinsic motivators, and social motivators.

Although a plethora of research can be found on crowds' motivation, however, we have not yet reached a consensus on the classification of the motivational factors. For instance, [27] considered community identification is an intrinsic motivation whereas [29] considered the desire for recognition or visibility as extrinsic motivation. Moreover, which motivational factor is more important for participants (crowd workers) is another subject to be investigated. For instance, [23] argued that monetary rewards are an important motivation for participating as [36] identified that earning money is one of the major motivation for the participants at InnoCentive. However, [28] argue there is no significant correlation between motivation for monetary reward and participation.

B. MOTIVATIONAL CONSTRUCTS

From sustaining participants to revealing their motivation many studies have been conducted. A substantial part of past research is upon what motives intricate the crowd to participate in crowdsourcing tasks and to ascertain the influences of expected benefits (intrinsic or extrinsic) on crowds' participation [32]. For example, based on value theory [14] identified that hedonic value (enjoyment) with small money enhances the continuance of the participants. Similarly, [28] found that enjoyment (intrinsic) and recognition (extrinsic) enhances the participation in TaskCN (Chines popular crowdsourcing platform) case. In a study based on classic motivation theory, work motivation theory and open source software [27] identified community-based motivation, immediate payoffs, delayed payoffs, and social motivation have a strong effect on the time spent on the platform. However to test the model a paid survey on Mechanical Turk [37] was conducted where most of the participants were mainly from two countries, i.e., US and India. On the same line, based on their previous study [34] constructed a model and tested on TaskCN. Reporting that monetary reward and enjoyment increases participants intention to continue participating in micro tasks.

Later, drawn from job engagement theory and the theory of job design [38] have developed a framework, focused on Kaggle [12]. It is a platform for data modeling competitions where most of the participants are Ph.Ds. The results of an online survey showed that emotional and cognitive engagements are positively related to creativity. The greater the contestants' psychological involvement, the higher the creativity. In contrast, our intention is to uncover the motivating factors fostering competitive software crowd to participate in CCSD. Considering social exchange theory as the theoretical lens and context related literature [32] indicated that monetary reward, skill enhancement, work autonomy, enjoyment and trust influence the crowd participation. [32] also argued that cognitive effort has a negative effect on solvers participation and cost factors should be considered to encourage participation. In another study on Kaggle, [39] identified the triggers of creative efforts and support the premise that positive creative experiences lead to the increased contributed effort. They also suggested Kaggle to attract intrinsically motivated solvers and try to raise intrinsic motivation by creating an enjoyable environment for better competitions among intrinsically driven participants.

Most of the past studies are with some of their limitations such as many of them have focused micro tasks offering platforms, some have focused single platform where most of the participants were from specific geographic locations. Our study does not focus on different spectrum of crowds and platforms, e.g., famous for micro-tasks completion or participants competing in non-rewarded challenges e.g., Single Round Matches (SRM: a 1.5-hour algorithm competition of TopCoder). Instead our area of interest is only competitive crowdsource software development. Our target participants are only software developers of two types:

- Competitive software developers who participate in CCSD.
- Software developers who don't want to participate in CCSD.

III. STATE OF THE ART LITERATURE ON MOTIVATION OF SOFTWARE CROWD

This section provides explanation and articulates the differences that separate this study from the existing state of the art literature that studies ''motivation''.

To understand the factors behind the success of the Open Source Software (OSS), researchers [40]–[43] have investigated what motivates OSS developers to contribute their time and effort to the development of free software. OSS developers choose to participate in OSS project as long as the benefits exceed the cost [40]. The participants may improve their programming skills through the active peer review, in which through public archive faulty programming style and logic are communicated back to the original author [40], [43]. OSS community does not require only one type of motivation instead its a combination of intrinsic, extrinsic and community related intrinsic motivation [40]. Factors that promised future monetary rewards e.g., building human capital and self-marketing play significant role [43]. There are evidences, many open source developers are employed by the firms and are being paid to take the active part in the community to develop specific components of the software for the firm [43], [44].

However, CCSD is distinctively different from OOS; the key difference is the ownership of the outcome. CCSD relies on an open call format, where typically a project is divided into several mini-tasks and posted to a platform [45]. Then the globally distributed software crowd compete to provide the best solution for the given task and only one or two winners

are rewarded with the monetary reward and others get no monetary incentive. To the best of our knowledge, no study provides the understanding about what motivate CCSD community to participate where winning monetary rewards are crucial and loss of cognitive efforts are high. To this end, based on extensive literature review, we design a survey to explore the motivating factors of the CCSD community. This study is one of the first studies that presents the in depth investigation about CCSD community and provides sound empirical data about what are major motivational factors for CCSD community.

IV. RESEARCH METHODOLOGY

A. RESEARCH QUESTIONS

The objective of this study is to probe deep into the phenomenon of CCSD. Investigate, why some software developers working in the industry don't want to participate in such competitions. Calculate, how many existing registrants/ developers of TopCoder contributed their solutions and how frequently a software developer wins the competitions. Our main research questions are listed as below:

- RQ1: What are the major motivating factors for competitive software developers to participate in CCSD?
- RQ2: What is inhibiting industrial software developers to participate in CCSD?
- RQ3: Does Pareto principle hold in the CCSD i.e., whether 80% of the competitions are won by the 20% of the developers?

Research question RQ1 is raised to identify the motivational factors that spur participation of software developers in CCSD. It also provides understanding about what existing studies validate and what real practitioners aimed when participate in such competitions where loss of cognitive efforts are high. The empirical evidence produced by the study will help understand how to enhance and sustain the continuance of the participants.

In order to maximize the participation, it is important to recognize and understand the factors that inhibit software developers working in the industry. How they can consider participation in such competitions and what do they think about the competitive crowd. Research question RQ2 aims to identify such barriers, addressing them may attract more potential contributors.

The Pareto principle is also known as the 80-20 rule introduced by Vilfredo Pareto in 1935 [47]. The rule claims that 80% of the output is a direct result of 20% input. It is a decision-making technique that can be applied to a different business domain to ensure that the business process is functioning to its full potential. Research question RQ3 investigates whether Pareto principle holds true for CCSD. The ultimate goal of applying Pareto principle is twofold. First, the platform can pay attention to most active and successful developers only instead of trying to increase the number of potential developers. Second, to determine the areas that require more attention to achieve maximum potential of CCSD. That is, the platform can calculate the popularity of the contests and percentage of the software developer who do not make any essential contributions.

B. STUDY DESIGN

This section explains the study design, selected constructs from existing studies, data collection method, and analysis performed on data to measure the reliability of the collected data.

1) MEASURING CONSTRUCT

We adopted the measuring constructs from existing literature. To find the extant literature on measuring constructs, we set the searching criteria of a paper who title should meet the following search string and variations of it: (''Crowdsourcing intrinsic motivation'' OR ''Crowdsourcing extrinsic motivation'' OR ''Competitive crowds motivation'' OR ''Competitive crowdsourcing motivation'' OR ''Competitive crowdsourcing incentives'' OR ''Crowdsourcing software development motivation'' OR ''Software crowdsourcing motivation'' OR ''Crowdsourcing competitions'' OR ''Crowdsourcing software developers motivation'' OR ''Motivation of competitive software developers'' AND ''online contest participants motivation''). We use the online libraries for searching the research papers which include IEEE Xplore, Springer library, Elsevier Science direct, ACM library, ResearchGate and Google scholar. We select the papers published between 2008 to 2018. The initial searching process led 64 papers in total. After applying manual search in which we read the title, abstract and introduction of each paper. In total 44 papers were selected out which 24 were directly related to crowdsourcing participation. We analyzed each paper and finally selected 13 studies that measure the motivation of crowds participating in crowdsourcing platforms.

First, we determine the 10 motivational factors used in the studies that focused motivation of the crowds. Table 1 shows the motivational constructs used in the studies. Second, based on a pilot we adopted 6 constructs for our empirical study. The adopted constructs are *money*, *learning*, *social contacts*, *peer recognition*, *hobby* and *future investment*. Moreover, to sums up our inquiry, sub-questions under each construct are asked in the questionnaire. It's a common approach the researcher divides main questions into two or more less cognitively taxing sub-questions, where the respondent estimates frequencies for the sub-questions and researcher combine sub-questions to obtain a value for global category [48]. The characteristics of sub-questions are identical to the main constructs, for instance *earning immediate* money [26], [29], [34] is directly related to *money* construct. The 2^{nd} column of the Table 2 shows the shorted form of sub-questions against each construct.

2) QUESTIONNAIRE

We used a web-based survey for data collection. Surveys are a good instrumental tool for getting empirical descriptions about trends, attitude and/or opinions of the population in

TABLE 1. Major constructs from existing studies.

Exiting Studies	Community	Learning	Money	Social Contacts	Enjoyment	Recognition	Investment	Hobby	Self-efficacy
Hua [32]		✓	✓		✓	✓		✓	✓
Kaufmann [27]	✓	✓	✓	✓		✓	✓	✓	✓
Barbham [29]		✓	✓	✓		✓			✓
Namousi [30]		✓		✓	✓	✓	✓		
Sun et al [14]	✓	✓	✓		✓				✓
Jorge et all [33]	✓			✓	✓				✓
Barbham [31]	✓	✓	✓	✓	✓	✓		✓	
Barbham [23]		✓	✓	✓		✓	✓		
Tanja et all [46]		✓	✓	✓		✓	✓	✓	
Zhen et all [28]		✓	✓	✓		✓	✓	✓	
Alexande [43]		✓	✓	✓	✓	✓	✓	✓	
Joseph et all [40]		✓	✓	✓	✓	✓	✓	✓	
Guido et all [42]		✓	✓	✓		✓	✓	✓	
Current Study		✓	✓	✓		✓	✓	✓	

TABLE 2. Answers to sub-questions.

Constructs	Sub-questions	Not at all	A little bit	Neutral	Agree	Strongly Agree
Money	Immediate Money	27%	17%	24%	23%	9%
	Main source of living	61%	11%	11%	10%	7%
	Extra money	25%	23%	13%	24%	15%
	Easy money	47%	10%	20%	18%	5%
Peer Recognition	Recognized by other professionals	10%	10%	27%	28%	25%
	Frequent feedback boots confidence	5%	14%	21%	38%	22%
	Feedback improve my skills	1%	15%	10%	42%	32%
	Innovative skills	3%	10%	18%	40%	29%
Learning	Improve prog. Skills.	1%	10%	10%	27%	52%
	Observe other code to be creative	5%	3%	12%	39%	41%
	Plan own professional project	6%	16%	23%	27%	28%
	Produce related products	19%	17%	14%	29%	21%
Social Contacts	Build professional network	4%	17%	18%	49%	12%
	Committed with community	7%	14%	30%	35%	14%
	Seek social identification	10%	18%	24%	33%	15%
Hobby	Enjoy writing code with good money	5%	18%	15%	29%	31%
	Feel productive	1%	17%	18%	34%	30%
	Enjoy finding solutions	1%	9%	15%	33%	42%
	Good way to spend time	10%	17%	11%	38%	24%
Future Investment	Win or learn for future	5%	7%	11%	36%	41%
	Sale same products	32%	5%	29%	28%	6%
	Shape my portfolio for job	7%	18%	17%	44%	14%
	Productive to see others solutions	3%	15%	10%	47%	25%

the study [49]. Moreover, our targeted audience was from different geographical locations so we used the web-based survey to maximize coverage and participation.

The online questionnaire was designed using closed-ended questions. The first version of the questionnaire was piloted using software development communities found on different software crowdsourcing platforms and social websites. As the test was conducted, possible changes were observed. Based on the feedback received from the pilot questionnaire, the final questionnaire[1] is amended.

The questionnaire was divided into three main parts (1) demographic data (years of experiences, gender, country etc.), (2) motivational drivers (3) inhibiting factors. All items were measured on 5 Likert scales anchored from *Not at all* to *Strongly Agree*. Likert-type scales are fixed choice response designed to measure attitude or opinion [50], [51].

C. DATA COLLECTION

Data is collected using online Google Forms services that let us create a structured survey questionnaire. The Google Forms kept accepting responses for more than 4 months. The survey required 15 minutes for the competitive participants to answer all the questions. However, for non-participants, it required less than 10 minutes only because the number of questions is reduced.

Our targeted audience is rarely available on micro-tasks solving platforms like Amazon Mechanical Turk. Therefore, we adopted a different approach to outreach our targeted audience. We viewed numbers of frequent participants profiles of competitive software crowd from different crowdsourcing platforms including TopCoder, hackerearth, CodeForces, and Kaggle to get their email addresses (if any) from their profile web pages. Most of the competitor developers also use social networking and question answering platforms e.g, Quora[2]

[1] https://goo.gl/forms/Xt8sFTkeNdJF4f0X2

[2] www.quora.com

TABLE 3. Descriptive statistics.

Constructs	No. of Responses	Minimum	Maximum	Mean	Std. Deviation
Money	113	1.00	5.00	2.4668	1.12426
Peer Recognition	113	1.60	5.00	3.6726	0.97289
Learning	113	2.00	5.00	3.8363	0.85354
Social Contacts	113	1.90	5.00	3.7650	0.95550
Hobby	113	1.33	5.00	3.2360	0.92226
Future Investment	113	1.57	5.00	3.3274	0.88527
Valid N (listwise)	113				

TABLE 4. Group statistics.

Constructs	Gender	No.of Responses	Mean	Std. Deviation	Std. Error Mean
Money	Male	93	2.4543	1.16268	0.12056
	Female	20	2.5250	0.94903	0.21221
Peer Recongnition	Male	93	3.7581	0.96435	0.10000
	Female	20	3.2750	0.93506	0.20909
Learning	Male	93	3.9274	0.84102	0.08721
	Female	20	3.4125	0.79998	0.17888
Social Contacts	Male	93	3.2760	0.88140	0.09140
	Female	20	3.0500	1.09904	0.24575
Hobby	Male	93	3.8489	0.95237	0.09876
	Female	20	3.4750	0.89185	0.19942
Future Investment	Male	93	3.3835	0.88321	0.09158
	Female	20	3.0667	0.86923	0.19437

and Facebook.[3] While tracing their online footprint, we used these platforms as an instrument to outreach the potential participants. Each participant was directly contacted using available means and requested to fill up the questionnaire. More than 400 potential profiles were outreached. We managed to get 113 valid responses from competitive crowdsourcing software developers. Moreover, we requested software developers working in different software development companies (with the help of our colleagues, old classmates and students) who don't participate in CCSD to fill up the survey and managed to get 75 responses.

In order to perform Pareto analysis we extracted history data of Software Development Challenge from TopCoder using its available Application Programming Interface (API). We targeted TopCoder because it is one of the most popular CCSD platforms with more than 1.2 million global community of software developers.

D. DATA ANALYSIS

During the pilot study, the theoretical constructs and the related items were analyzed for the content validity and the related items were initially purified by corrected item-total correlation.

We calculated descriptive statistics which is used to explain the data set characteristics [52], Table 3 shows descriptive statistics that include measures of central tendency and measures of spread which enable us to present the data in a more meaningful way with simpler interpretation. For example, Mean is used to depict average response to a specific question i.e., central tendency of data. Standard deviation is a measure of spread that informs if the variables deviate from central location. In our data sample the *money* construct has lowest mean and *learning* construct has highest mean score. Additionally, gender based group statistics of respondents are also presented in Table 4.

We performed Exploratory Factor Analysis (EFA) to check unidimensionality with varimax rotation along with extraction by principal components in SPSS. Exploratory Factory Analysis (EFA) is a method used to determine the underlying relationships between measured variables. It is used to identify the structure of the relationship between the variable and the respondent. It tells importance of each construct in the overall instrument. If factor loading of any parameter is less than 0.6, we can drop or change that parameter in the instrument to find significant relationship among variables. EFA is commonly used by the researchers for developing questions that measure a particular research topic [53]. Table 5 shows factor loading of each sub-question and coronach alpha value of all constructs.

In order to determine the reliability of the scale, we used Cronbach's alpha [54] technique. It is a well-known and most commonly used technique of internal consistency. Cronbach's Alpha determines the extent to which variables are positively related to each other for multiple Likert questions in a questionnaire. Therefore, the Cronbach's Alpha reliability test was applied using SPSS statistics tool and the

[3] www.facebook.com

TABLE 5. Statistics of explanatory factor analysis.

Constructs	Items	Factor Loading	Cronbach-α
Money	How much it important for you to earn money through competitions? [Comparing to other programming job, winning through competitions give immediate money.]	0.845	0.850
	How much it important for you to earn money through competitions? [It is my main source to earn my living]	0.701	
	How much it important for you to earn money through competitions? [To earn some extra money]	0.933	
	How much it important for you to earn money through competitions? [I consider it easy way to earn good money]	0.834	
Peer Recognition	How much Peer Recognition is important for you? [To be well recognized by other professionals]	0.730	0.865
	How much Peer Recognition is important for you? [Frequent feedback boost my confidence]	0.855	
	How much Peer Recognition is important for you? [Authentic feedback improve my skills]	0.922	
	How much Peer Recognition is important for you? [It foster my innovative skills]	0.876	
Learning Intentions	What are you learning intentions ? [To improve my programming skills]	0.672	0.713
	What are you learning intentions? [By observing other's solutions I can be more creative]	0.644	
	What are you learning intentions? [It help me plan my own professional project]	0.899	
	What are your learning intentions? [To produce related products in the marketplace]	0.715	
Social Contacts	How much Social Contacts are important for you when participating? [To build a professional network of software developers]	0.880	0.799
	How much Social Contacts are important for you when participating? [I feel committed with other community members]	0.919	
	How much Social Contacts are important for you when participating? [I seek social identification by participating]	0.750	
Hobby	How much Hobby is important for you as a motivator factor when you participate? [I enjoy writing code and its fun for me with good money]	0.815	0.857
	How much Hobby is important for you as a motivator factor when you participate? [I feel more productive and satisfied when I participate]	0.880	
	How much Hobby is important for you as a motivator factor when you participate? [I enjoy finding solutions of complex problems]	0.835	
	How much Hobby is important for you as a motivator factor when you participate? [Its a good way to spend my free time]	0.837	
Future Investment	How much Future Investment is important for you as a motivation for participating? [I can sale same type of services and products in the marketplace]	0.721	0.649
	How much Future Investment is important for your as a motivation for participating? [I shape my portfolio in order to get a better job]	0.809	
	How much Future Investment is important for you as a motivation for participating? [I can be more creative to see others professional solutions]	0.677	

* We drop the first sub-question of Future Investment construct from EFA because of weak factor loading value.

obtained alpha value is 0.872 which can be seen as satisfactory for our application. If the alpha value is less than 0.70 than it is considered questionable and unacceptable [55]. In our case, construct validity is not necessary because we adopted the constructs from extant literature shown in Table 1.

Moreover, the one-way analysis of variance (ANOVA) is performed to determine the statistically significant differences between the mean of demographically different population. Our pilot study revealed that qualification, income and occupation are important variables which create differences in responses. For example, 5th row of Table 7 shows that for a person whose annual income is between $30,000 - $36,000 social contact is more important motivation whereas for a person whose earning is between &1000 - $12,000 learning is more important motivation. One possible reason for such difference is the varying income level of different countries. For instance, per capita income of India is significantly different than USA. The Anova results show, in our case Qualification, Income and Occupation lead significant differences among the motivating factors. Tables 6, 7 and 8 show the oneway ANOVA test results of Qualification vs. Motivation Factors, Income vs. Motivation Factors and Occupation vs. Motivation Factors respectively.

V. RESULTS

This section presents the results of the participants' demographic and answers to the stated research questions.

A. DEMOGRAPHIC

In total 193 professionals participated in the web-based questionnaire where 75 of them are non-participants who don't participate in crowdsourcing software competitions. After performing manual validation on collected data, we pruned out 11 responses, 6 from non-participants professionals and 5 from participants professionals.

Demographically, among nonparticipants (40%) respondents fall in the age bracket of 15-25 years and (46%) in 25-35 years. The collected data show (10%) respondents hold PhD degree, (33%) respondents hold master degree and

TABLE 6. Oneway ANOVA qualification vs motivational factors.

Constructs	Qualification	No. of Responses	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
						Lower Bound	Upper Bound		
Money	High school graduate	23	2.1196	0.92892	0.19369	1.7179	2.5213	1.00	4.25
	College graduate	50	2.2050	1.05426	0.14909	1.9054	2.5046	1.00	4.75
	Master	40	2.9938	1.14142	0.18047	2.6287	3.3588	1.00	5.00
	Total	113	2.4668	1.12426	0.10576	2.2573	2.6764	1.00	5.00
Peer Recognition	High school graduate	23	3.3152	1.20891	0.25208	2.7924	3.8380	1.50	5.00
	College graduate	50	3.6400	0.93016	0.13154	3.3757	3.9043	2.00	5.00
	Master	40	3.9188	0.81726	0.12922	3.6574	4.1801	1.75	5.00
	Total	113	3.6726	0.97289	0.09152	3.4912	3.8539	1.50	5.00
Learning	High school graduate	23	3.7717	0.90111	0.18790	3.3821	4.1614	2.00	5.00
	College graduate	50	3.9100	0.95131	0.13454	3.6396	4.1804	2.00	5.00
	Master	40	3.7813	0.69611	0.11007	3.5586	4.0039	2.75	5.00
	Total	113	3.8363	0.85354	0.08029	3.6772	3.9954	2.00	5.00
Social Contacts	High school graduate	23	3.2029	0.99868	0.20824	2.7710	3.6348	1.67	5.00
	College graduate	50	3.0400	0.91260	0.12906	2.7806	3.2994	1.33	4.67
	Master	40	3.5000	0.84395	0.13344	3.2301	3.7699	1.67	5.00
	Total	113	3.2360	0.92226	0.08676	3.0641	3.4079	1.33	5.00
Hobby	High school graduate	23	3.8152	1.06392	0.22184	3.3551	4.2753	2.00	5.00
	College graduate	50	3.8300	0.98644	0.13950	3.5497	4.1103	2.00	5.00
	Master	40	3.9375	0.86741	0.13715	3.6601	4.2149	2.00	5.00
	Total	113	3.8650	0.95550	0.08989	3.6869	4.0431	2.00	5.00
Future Investment	High school graduate	23	3.4928	0.83406	0.17391	3.1321	3.8534	1.67	5.00
	College graduate	50	3.2867	0.91847	0.12989	3.0256	3.5477	1.67	4.67
	Master	40	3.2833	0.88208	0.13947	3.0012	3.5654	1.67	5.00
	Total	113	3.3274	0.88527	0.08328	3.1624	3.4924	1.67	5.00

(30%) respondents are college graduate. For the participants, (46%) respondents fall in the age bracket of 15-25 years and (41%) in 25-35 years. Majority of the respondents hold the graduate degree (44%) and Master degree holders are (35%). Most of the participants have regular jobs status (46%) followed by *students* which is almost (25%).

Among nonparticipants, almost (34%) respondents have less than 1 year of experience and (24%) respondents have 1 to 3 years of experience. Rest of the respondents have more than 3 years of experience in the software industry. Figure 1 shows the software development experience of nonparticipants. On the x-axis, it shows the number of participants and on the y-axis, it shows years of experience. Whereas, for the participants' majority of the respondents (55%) have more than 3 years of experience in software industry and (17%) of the respondents have less than a year of experience. Figure 2 shows the software development experience of participants who take part in CCSD.

Table 9 illustrates (77%) respondents of the participants won the competitions more than 1 times and have been among top five positions in crowdsourcing competitions. The first column of the Table 9 shows the percentage of participants and the second column shows the range of wins. Most of the respondents (27%) won the competitions 1 to 5 times. There are about (12%) participants who won 5 to 10 times, (16%) won 10 to 50 times, (7%) won 50 to 100 times, (2%) won 100 to 200 times and (2%) won the competitions more than 200 times.

B. RQ1: MOTIVATING FACTORS

Through an online survey, 116 crowdsource software development competitors responded whereas 5 responses were disregarded due to missing values. In total 113 responses were considered and analyzed.

1) THE MOST RATED MOTIVATIONAL FACTOR

To investigate the question "What motivate you the most to participate in CCSD?", six motivational constructs (*money*, *peer recognition*, *learning*, *social contact*, *hobby* and *future investment*) are given to the respondents to select their most preferred choice for CCSD participation.

The results are summarized in Table 10 that clearly shows, for most of the participants, *learning* is a most preferred choice that motivates them to participate in CCSD. Followed by *social contacts* and *peer recognition*. Surprisingly, majority of the respondents gave least preference to the *money* construct (as shown in Table 10) as compared with *learning* construct.

From the 4^{th} row of the Table 10 it is observable that (58% accounting for 65/113) of the participants strongly agree and (29% accounting for 33/113) are agree that *learning* is the major motivational factor to participate in CCSD. One possible reason for such a tendency could be a collaborative environment provided by the platforms such as TopCoder, where competitors give tips to their fellow competitors. Besides that, software solutions provided by the competitors need to go through from certain phases where direct feedback help improve quality and skills of participants. For instance TopCoder review process, where both reviewers and submitter go through lively discussion and feedback mechanisms in order to achieve higher quality output and improved skill.

Table 2 presents results of sub-questions asked for each motivational construct. It shows that more than half of the participants put their effort into improving programming skills. It also shows that about half of the participants have the intention to observe others' code in order to improve their creative skills. For instance in answering the question "What are your learning intentions?"; most of the participants think it is a good way to improve their programming skills.

TABLE 7. Oneway ANOVA income vs. motivational factors.

Constructs	Income	No. of Responses	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
						Lower Bound	Upper Bound		
Money	$1000	15	2.2333	1.20811	0.31193	1.5643	2.9024	1.00	4.75
	$1000-$12,000	17	2.0588	1.05893	0.25683	1.5144	2.6033	1.00	3.75
	$12,000-$18000	17	2.5000	0.96825	0.23483	2.0022	2.9978	1.00	4.25
	$18,000-$24,000	14	2.3571	1.05936	0.28313	1.7455	2.9688	1.00	4.25
	$24,000-$30,000	13	2.4038	1.12054	0.31078	1.7267	3.0810	1.00	4.25
	$30,000-$36,000	3	3.4167	1.01036	0.58333	0.9068	5.9265	2.25	4.00
	$36,000 and more	34	2.7426	1.19735	0.20534	2.3249	3.1604	1.00	5.00
	Total	113	2.4668	1.12426	0.10576	2.2573	2.6764	1.00	5.00
Peer Recognition	$1000	15	3.5167	1.16675	0.30125	2.8705	4.1628	1.50	5.00
	$1000-$12,000	17	3.5882	0.76004	0.18434	3.1975	3.9790	2.00	4.75
	$12,000-$18000	17	3.2647	1.02877	0.24951	2.7358	3.7936	1.75	5.00
	$18,000-$24,000	14	3.8571	1.15073	0.30755	3.1927	4.5216	1.75	5.00
	$24,000-$30,000	13	3.8654	1.11157	0.30829	3.1937	4.5371	2.00	5.00
	$30,000-$36,000	3	3.0000	1.25000	0.72169	-0.1052	6.1052	1.75	4.25
	$36,000 and more	34	3.8971	0.74396	0.12759	3.6375	4.1566	2.25	5.00
	Total	113	3.6726	0.97289	0.09152	3.4912	3.8539	1.50	5.00
Learning	$1000	15	3.7833	0.83381	0.21529	3.3216	4.2451	2.75	5.00
	$1000-$12,000	17	3.9118	0.72317	0.17540	3.5399	4.2836	3.00	5.00
	$12,000-$18000	17	3.8676	0.89319	0.21663	3.4084	4.3269	2.25	5.00
	$18,000-$24,000	14	4.0179	0.93780	0.25064	3.4764	4.5593	2.00	5.00
	$24,000-$30,000	13	3.3846	0.98750	0.27388	2.7879	3.9814	2.00	5.00
	$30,000-$36,000	3	4.1667	0.62915	0.36324	2.6038	5.7296	3.50	4.75
	$36,000 and more	34	3.8750	0.83768	0.14366	3.5827	4.1673	2.00	5.00
	Total	113	3.8363	0.85354	0.08029	3.6772	3.9954	2.00	5.00
Social Contacts	$1000	15	3.0444	0.82488	0.21298	2.5876	3.5012	2.00	4.33
	$1000-$12,000	17	3.1569	0.65741	0.15945	2.8189	3.4949	2.00	4.33
	$12,000-$18000	17	3.0000	0.96465	0.23396	2.5040	3.4960	1.67	5.00
	$18,000-$24,000	14	3.2857	1.29995	0.34743	2.5351	4.0363	1.33	5.00
	$24,000-$30,000	13	3.3590	1.10940	0.30769	2.6886	4.0294	1.33	5.00
	$30,000-$36,000	3	4.0000	0.33333	0.19245	3.1720	4.8280	3.67	4.33
	$36,000 and more	34	3.3431	0.83479	0.14316	3.0519	3.6344	1.67	4.67
	Total	113	3.2360	0.92226	0.08676	3.0641	3.4079	1.33	5.00
Hobby	$1000	15	3.9000	0.95805	0.24737	3.3695	4.4305	2.25	5.00
	$1000-$12,000	17	4.0735	0.92180	0.22357	3.5996	4.5475	2.25	5.00
	$12,000-$18000	17	3.5147	1.00573	0.24392	2.9976	4.0318	2.25	5.00
	$18,000-$24,000	14	3.9821	0.93266	0.24926	3.4436	4.5206	2.00	5.00
	$24,000-$30,000	13	3.6923	1.24647	0.34571	2.9391	4.4455	2.00	5.00
	$30,000-$36,000	3	3.3333	0.94648	0.54645	0.9821	5.6845	2.25	4.00
	$36,000 and more	34	3.9853	0.83698	0.14354	3.6933	4.2773	2.00	5.00
	Total	113	3.8650	0.95550	0.08989	3.6869	4.0431	2.00	5.00
Future Investment	$1000	15	3.0667	0.90150	0.23277	2.5674	3.5659	1.67	4.00
	$1000-$12,000	17	3.6667	0.65617	0.15914	3.3293	4.0040	2.00	4.33
	$12,000-$18000	17	3.6471	1.01701	0.24666	3.1242	4.1700	1.67	5.00
	$18,000-$24,000	14	3.3571	0.99111	0.26488	2.7849	3.9294	1.67	5.00
	$24,000-$30,000	13	3.3077	0.84395	0.23407	2.7977	3.8177	1.67	4.33
	$30,000-$36,000	3	4.1111	0.19245	0.11111	3.6330	4.5892	4.00	4.33
	$36,000 and more	34	3.0392	0.81964	0.14057	2.7532	3.3252	1.67	4.33
	Total	113	3.3274	0.88527	0.08328	3.1624	3.4924	1.67	5.00

The results of sub-questions are summarized in Table 2, where the 4^{th} row of the Table 2 shows (46% accounting for 52/113) participants are strongly agree and (23% accounting for 27/113) are agree that improving programming skills is their main learning intention. Likewise, (36% accounting for 41/113) strongly agree and (35% accounting for 39/113) are agree that observing others' code can make them more creative. Moreover, (24% accounting for 27/113) strongly agree and (25% accounting for 28/113) are agree that it helps them plan their own professional projects. Almost (26% accounting for 29/113) strongly agree and (19% accounting for 21/113) are agree, by participating they could produce the related products in the marketplace.

2) DIRECT ECONOMIC REWARD

Table 3 shows that *money* construct has the lowest mean value that indicates majority of the respondents have given low rating to the *money* construct, moreover, the results of oneway ANOVA test shown in Tables 6, 7 and 8 also suggest that *money* construct has lowest importance for demographically different population in the collected sample.

Our research findings differ from [23] and [36] who argue, possibility of earning money is a dominant motivation. Instead in our case *learning* is a dominant motivation for software developers participating in the CCSD. However, our findings are more aligned with a study conducted in 2002 on the motivation of open source developers [43], where (79.9%) respondents chose *Improving my programming skills* when they asked why they participated in open-source projects. Similarly in our case, as Figure 3 shows, (51%) respondents chose *Yes because I learn more programming skills when competing* when they asked will you participate if the challenges posted on crowdsource platforms would have no monetary benefits.

C. RQ2: INHIBITING FACTORS

In total 75 non-participants professional (who don't take part in CCSD) participated in the web-based questionnaire from which 6 responses were discarded during manual validation check.

1) MOST RATED INHIBITING FACTORS

Table 11 shows that almost (38%) of the non-participants professionals respondents, who have never participated in crowdsource competitions, think that they don't have the

TABLE 8. Oneway ANOVA occupation vs. motivational factors.

Constructs	Occupation	No. of Responses	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
						Lower Bound	Upper Bound		
Money	Regular job holders	53	2.5189	1.17756	0.16175	2.1943	2.8434	1.00	5.00
	Unemployed	1	2.0000	.	.	.	.	2.00	2.00
	Self-employed	17	2.7500	1.08613	0.26343	2.1916	3.3084	1.00	4.00
	Student	29	2.3362	1.09641	0.20360	1.9192	2.7533	1.00	4.75
	Freelancer	13	2.2115	1.07455	0.29803	1.5622	2.8609	1.00	3.75
	Total	113	2.4668	1.12426	0.10576	2.2573	2.6764	1.00	5.00
Peer Recognition	Regular job holders	53	3.7264	0.88288	0.12127	3.4831	3.9698	1.75	5.00
	Unemployed	1	2.0000	.	.	.	.	2.00	2.00
	Self-employed	17	3.9118	0.87026	0.21107	3.4643	4.3592	2.50	5.00
	Student	29	3.5690	1.08746	0.20194	3.1553	3.9826	1.50	5.00
	Freelancer	13	3.5000	1.14109	0.31648	2.8104	4.1896	2.00	5.00
	Total	113	3.6726	0.97289	0.09152	3.4912	3.8539	1.50	5.00
Learning	Regular job holders	53	3.7783	0.81551	0.11202	3.5535	4.0031	2.00	5.00
	Unemployed	1	2.0000	.	.	.	.	2.00	2.00
	Self-employed	17	4.0441	0.58118	0.14096	3.7453	4.3429	3.00	5.00
	Student	29	3.9397	0.84951	0.15775	3.6165	4.2628	2.75	5.00
	Freelancer	13	3.7115	1.17636	0.32626	3.0007	4.4224	2.00	5.00
	Total	113	3.8363	0.85354	0.08029	3.6772	3.9954	2.00	5.00
Social Contacts	Regular job holders	53	3.2956	0.83118	0.11417	3.0665	3.5247	1.67	5.00
	Unemployed	1	1.3333	.	.	.	.	1.33	1.33
	Self-employed	17	3.3529	0.98932	0.23995	2.8443	3.8616	2.00	5.00
	Student	29	3.2299	0.93479	0.17359	2.8743	3.5855	1.67	5.00
	Freelancer	13	3.0000	1.09713	0.30429	2.3370	3.6630	1.33	4.67
	Total	113	3.2360	0.92226	0.08676	3.0641	3.4079	1.33	5.00
Hobby	Regular job holders	53	3.8208	0.90455	0.12425	3.5714	4.0701	2.00	5.00
	Unemployed	1	2.0000	.	.	.	.	2.00	2.00
	Self-employed	17	4.2353	0.57602	0.13971	3.9391	4.5315	3.00	5.00
	Student	29	3.8448	1.05943	0.19673	3.4418	4.2478	2.00	5.00
	Freelancer	13	3.7500	1.18585	0.32890	3.0334	4.4666	2.00	5.00
	Total	113	3.8650	0.95550	0.08989	3.6869	4.0431	2.00	5.00
Future Investment	Regular job holders	53	3.2704	0.86493	0.11881	3.0320	3.5088	1.67	5.00
	Unemployed	1	1.6667	.	.	.	.	1.67	1.67
	Self-employed	17	3.3922	0.98061	0.23783	2.8880	3.8963	1.67	4.67
	Student	29	3.5057	0.81465	0.15128	3.1959	3.8156	1.67	5.00
	Freelancer	13	3.2051	0.94808	0.26295	2.6322	3.7780	1.67	4.33
	Total	113	3.3274	0.88527	0.08328	3.1624	3.4924	1.67	5.00

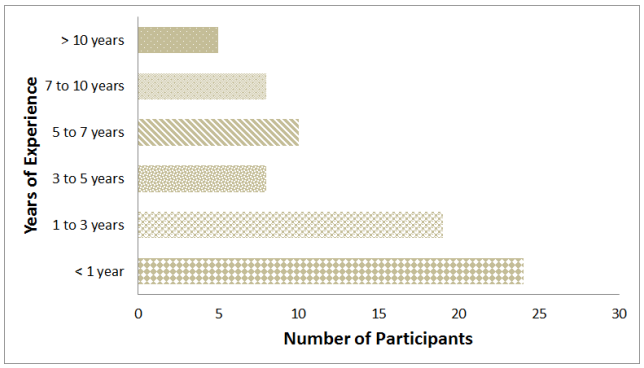

FIGURE 1. Software development experience of non-participants.

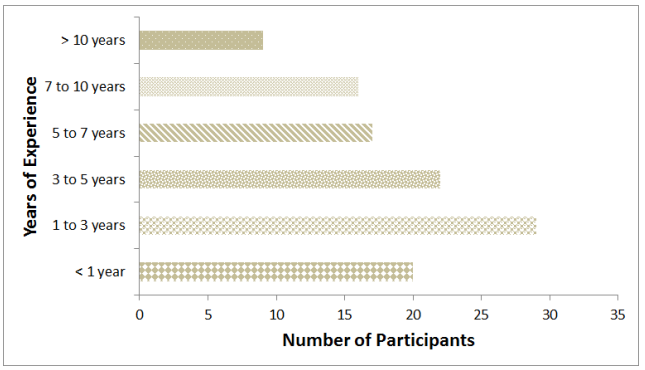

FIGURE 2. Software development experience of participants.

required competitive expertise to contest in the software development competitions. This indicates that industrial practitioners may think there are thousands of competitors in the marketplace, where, only the top one, two or three developers will win the monetary reward. Therefore, participation in such competition, only for monetary rewards, is less attractive. Moreover, (35%) of the respondents spend most of their time doing their professional jobs. This indicates that they prefer regular job over crowdsourcing tasks completion. Out of the total participants, (15%) prefer freelancing over crowdsourcing for earning money and (6%) think they would receive no compensation for their work. There were only (6%) participants who are not well aware of what is crowdsourcing.

TABLE 9. Crowdsource experience of respondents.

No. of wins	Percentage of Respondents
0	26%
1 to 5	31%
5 to 10	13%
10 to 50	18%
50 to 100	8%
100 to 200	2%
>200	2%

A question about potential participation in the future is asked (shown in Table 12) to the non-participants and 4 options were given to choose. Most of the non-participants chose that they can consider participation if it could shape

TABLE 10. Participants' rating for each motivational factor.

Constructs	Not at all	A little bit	Neutral	Agree	Strongly agree
Money	25%	19%	23%	19%	14%
Peer Recognition	5%	10%	26%	42%	17%
Learning	2%	5%	6%	29%	58%
Social Contacts	6%	14%	20%	42%	18%
Hobby	10%	13%	19%	30%	28%
Future Investment	6%	14%	23%	29%	28%

TABLE 11. Rating for inhibiting factors.

Inhibiting factors	Percentages
I don't have expertise required to contest in the software development competitions.	38%
I have no time because of my professional job otherwise I would participate.	35%
I prefer earning money through other freelancing/overtime means.	15%
I would receive no compensation for my work.	6%
I am not aware about crowdsource.	6%

TABLE 12. Potential participation in future.

Consider participation	Not at all	A little bit	Neutral	Agree	Strongly agree
If challenges give high payments.	19%	37%	16%	26%	7%
If the challenges match my expertise.	17%	19%	19%	31%	14%
If challenges have not limited time.	27%	17%	32%	20%	4%
If it could shape my professional growth.	11%	19%	12%	22%	36%

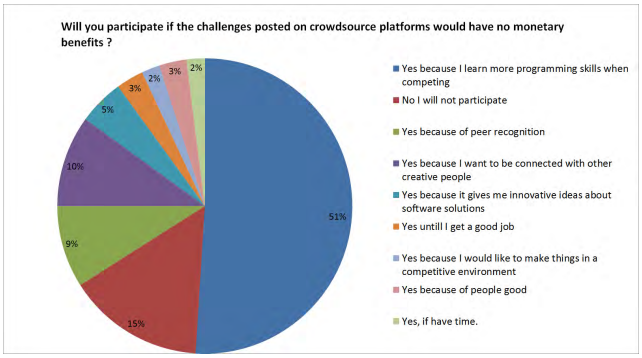

FIGURE 3. Consideration of participants without monetary reward.

their professional growth. The analysis suggests, more than half of the professional practitioners preferred career development over monetary rewards. The results are summarized in Table 12, that shows a significant amount of non-participants (14%) are strongly agree and (31%) are agree, if challenges match their expertise they can consider participation in crowdsourcing competitions. This indicates, some industrial practitioners perceive that only expert programmers can participate in such competitions. Moreover, from the results, the monetary attraction is also found to be little attractive as (19%) chose *Not at all* and (32%) considered it *A little bit*.

2) OPINIONS ABOUT COMPETITIVE DEVELOPERS

To understand what do non-competitive software developers think about competitive software developers a question ''Your opinion about the people participating in Crowdsource Software Development Competitions?'' was asked.

Figure 4 represents the results, most of the non-participants professionals (29%) think those who participate in such competitions want to improve their programming skills. Almost same amount of the respondents (27%) think they are expert programmers who can produce the quality solution within the strict time limit. A small portion of respondents (7%) think they don't have a good job and few are in an opinion that they want to get famous among software developer community. About (12%) has no opinion and (12%) think by participating they make their portfolio better and that may help them in securing a good job.

D. RQ3: PARETO PRINCIPLE

The Pareto analysis is a business decision making technique that statistically separate the factors having greatest impact on outcomes, either desirable or undesirable. The principle is based on the facts that 80% of the effects come from 20% of the causes or conversely 80% of problems are traced to 20% of the causes. The rule has been applied in different fields ranging from Criminology, Biology, Business and Economics. In software engineering Microsoft observed that 80% of the bugs would be eliminated by fixing the top 20% of the most reported bugs [56]. In order to apply the Pareto analysis, we extracted the history data from TopCoder using its available API, for our study, we extracted the data of only past Software Development Challenges results using a script written in C sharp. The data was analyzed using python language, our results of the analysis of the history data suggest that Pareto principle holds for CCSD.

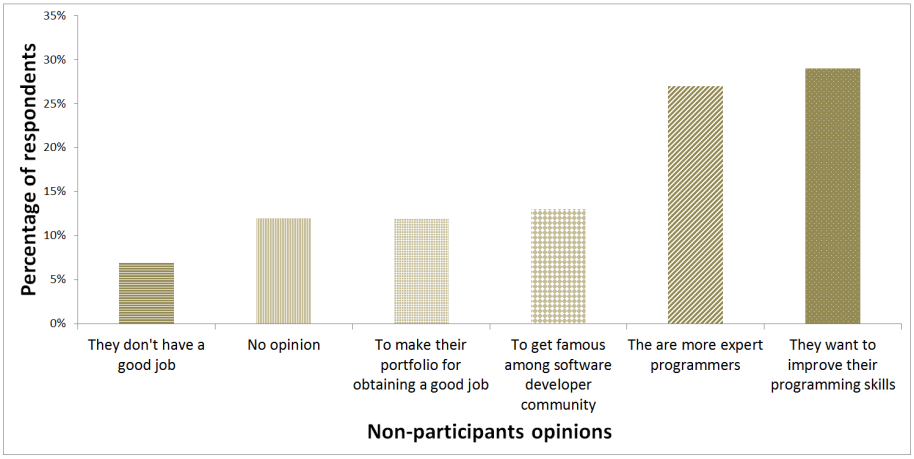

FIGURE 4. Non-participants professionals opinion about participants professionals.

TABLE 13. Statistical matrix of TopCoder.

Challenges	No. of software developers
Total Challenges	15287
Total Registrants	268872
Total Submissions	45101
Zero Submission	1594
Failed Submission	1691
Challenges remain	13151
First Placements	13151
Second Placements	8778
Third Placements	5180

1) QUICK FACTS AND STATS

The basic statistics of collected data are shown in Table 13 which indicates, for the total 15287 past software development challenges there are 268872 numbers of registrants who registered for the 15287 software development challenges. However, only 45101 submissions are made which is almost (17%) of the total number of registrants. Alternately, on average there are only 3 submissions for each competition. It is observed that, there are 1594 software development challenges who failed to meet a single submission which is (10%) of the total software development challenges. For the 542 software challenges, which is (4%) of 15287, there are 1691 submissions that failed either screening tests or review scoring.

Precisely there remain 13151 software challenges, which is (86%) of 15287, who have successful submissions and winners. However, not all (86%) challenges have three placements (first, second and third winners). It is found that there are 8778, which is almost (67%) of 13151, challenges that have first and second placements. And 5180 challenges, which is (39%) of 13151, have third placement as well.

2) 0.9% OF THE REGISTRANTS WON 86% SOFTWARE CHALLENGES

It is observed that, till July 2017, there are total 2418 unique software developers who won all the competitions.

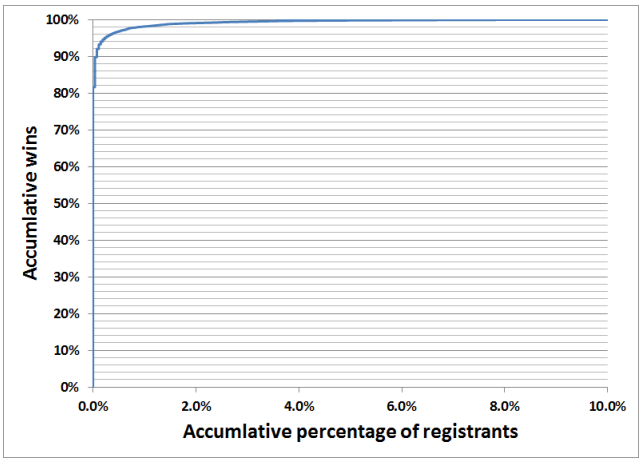

FIGURE 5. Pareto line.

We applied the Pareto analysis on collected data and created cumulative chart shown Figure 5. It is found that there are only (0.9%) software developers who are winning all the competitions. That means (0.9%) of total 268872 registrants have won 13151 software development challenges with 45101 submissions.

Out of (0.9%) competitors, there are almost (54%) competitors who win most of the competitions. Figure 6 shows the Pareto distribution of (0.9%) competitors. We can see from the Figure 6 small numbers of registrants win most of the competitions.

The long tail graph shown in Figure 7 revealed that the highest wins a registrant has is above 450 times. We categorized the winners into five separate groups shown in Table 14. For the first group, there are 1927 winners who won the competitions for 1 to 10 times. It is important to mention here, there are 1074 software developers, which is (56%) of 1927, who won only 1 times. Similarly, in second and third groups, there are 361 and 81 software developers who won the competitions 11 to 50 times and 51 to 100 times respectively. In the fourth group, 37 software developers are found who

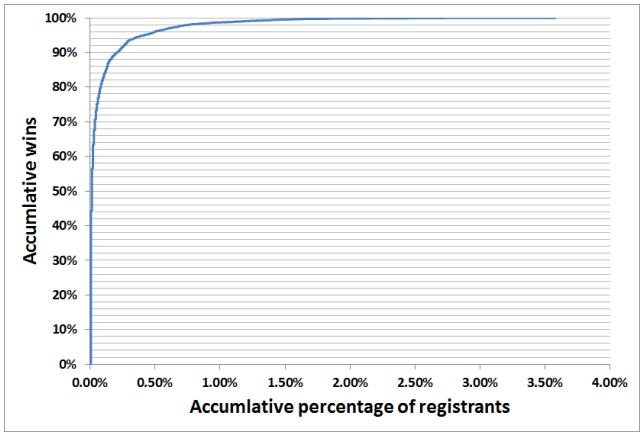

FIGURE 6. Pareto line (top 4% developers).

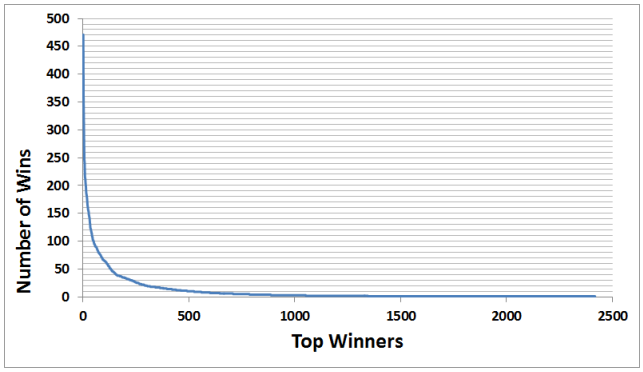

FIGURE 7. Frequency of wins for top winners.

TABLE 14. How often winners won.

Wins	No of winners
1 to 10	1927
11 to 50	361
51 to 100	81
101 to 200	37
201 to 300	9
301 to 500	3

won 101 to 200 times. In the group five, 9 software developers are found who have won for 201 to 300 times. And there are 3 software developers who won 301 to 500 times, precisely these 3 winners have won 471 times followed by 364 times and 331 times. Driven by the facts, we can say that (86%) software developments challenges are won by (0.9%) of the registrants.

The Pareto analysis help conclude the problem faced by the TopCoder i.e., most of the contributions are made by the small numbers of software developers that may lead to less number of submissions, poor quality of the complex software solutions and dissatisfaction of the customers indicated in different studies [17]–[19], [57]. The Pareto analysis suggest, in TopCoder case out of more than 1.2 million software developers crowd only few take it as serious job. The Pareto analysis does not provide the solution to the problem, instead it would help platform to create strategies and plans address the problem.

VI. IMPLICATION

Following subsections provide the detailed discussion of the results in perspective of crowdsourcers, crowdsourcing platforms, and researchers.

A. CROWDSOURCERS

Our findings suggest that crowdsourcing software development is still in a very early stage, and thus we suggest crowdsourcers post simple (but maybe time-consuming) tasks instead of highly challenging ones that may require rich experience and advanced technologies. The analysis in Section V-B.1 suggests that most of the software engineers participate in crowdsourcing software development for career development, e.g., learning and reputation. Few of them take it as a serious job. All of these indicate that the majority of the participants are students or junior software developers, and few senior and productive software engineers participant in crowdsourcing software development. As a result, as revealed in Section V-D.1, many crowdsourcing tasks, especially complex tasks, fail to receive any satisfying submissions. The findings in Section V-C.2 also suggest that increasing the monetary rewards may not increase significantly the number of satisfying solutions. Consequently, to increase the likelihood of receiving good solutions by crowdsourcing, crowdsourcers should post tasks that could be resolved successfully by inexperienced developers (or even students).

B. PLATFORMS

Crowdsourcing platforms like TopCoder should attract participants by awarding them reputation (instead of money) and shaping their career. The findings in Section V-B.1 suggest that most of the participants take learning and reputation as their major motivation for crowdsourcing development. Besides that, the analysis of inhibiting factoring in Section V-C.1 suggests that increasing monetary reward may fail to increase the number of participants significantly. Consequently, to attract and keep a large number of participants, crowdsourcing platforms should be more helpful for participants' career development.

Platforms should pay special attention to a small part of the participants who are significantly more productive than others. Our findings in Section V-D.2 suggest that out of over 1.2 million software crowd in TopCoder, 2418 competitors won all of the 15286 software development challenges. Consequently, to improve the likelihood of receiving satisfying solutions, crowdsourcing platforms should try their best to attract such productive competitors, and post tasks that they are really interested in. In contrast, increasing the number of participants may not be the best strategy for success.

C. RESEARCHERS

CCSD is still in the early state, and there are many critical research issues to be resolved by researchers. The first issue that deserves researchers' attention is how to attract professional and senior software developers. For instance on TopCoder task quieting rate is about (87.4%) [17]. Different from other crowdsourcing tasks (e.g., classification of pictures), software development is much more complex, and thus it requires much more effort and skills. Consequently, before we can post complex software development tasks, we should attract a large number of professional and senior developers.

The second issue that deserves researchers' attention is how to predicate the likelihood of a given task to receive satisfying solutions. Our findings in Section V-D.1 suggest that tasks often fail to receive satisfying solutions. Knowing in advance whether the task can receive satisfying solutions help crowdsourcers to plan in advance and thus to reduce the risk.

VII. THREAT TO VALIDITY

The first threat to construct validity is the contexts of motivational factors (e.g., learning and money) are general and may not have clear boundaries. For instance, the participants may have a different understanding of the *learning* construct. To minimize the threat, first, we finalized 6 out of 10 motivational factors for the study that are from extant literature shown in Table 1. Second, we designed sub-questions shown in Table 2 to define the boundaries of each motivational factor.

The second threat to construct validity is that collecting feedback online through questionnaires may introduce some bias. Face-to-face interviews help with more accurate screening, e.g., capturing verbal and non-verbal [58], [59]. In contrast, the data collected with online questionnaire could be inaccurate and misleading. Apart from the fact, we selected an online questionnaire for the data collection of our study because our target audience is not accessible either for a face-to-face meeting or online meeting due to the geographical distance. To minimize the threat, we conducted the pilot study and inspected common method bias which examined the measurement context effect using Harmans single factor test.

The first threat to internal validity is the authenticity of the participants. Our targeted audience were software developers who either participate in CCSD or work in the software industry. To collect the samples, we focused on crowdsourcing platforms (e.g., TopCoder, Kaggle, Hackerearth, Quora, and other social networks). These platforms are suitable to find representative participants. However, fake profiles or multiple accounts of participants may lead to incorrect responses. Therefore, we scrutinized the participants belong to crowdsourcing software development or software industry, and directly contacted them through emails.

The first threat to external validity is that the number of responses is limited. We expected to have more than 300 responses. However, we received 113 and 75 responses from CCSD participants and non-participants respectively. A low response rate is not good for rigorous statistical analysis [60]. One possible reason for such a low response rate is that we did not conduct the paid survey. We directly contacted and requested the participants to fill up the questionnaire. However, besides several requests, the expected response rate was not achieved. The same study with a high response rate of the online questionnaire participants may variate the questionnaire-based results. To confirm the questionnaire-based conclusions, we conducted the history data analysis whose results (as shown in Section V-D.2) confirm the questionnaire-based conclusions.

The second threat to external validity is that the history data is limited to the developers from TopCoder crowdsource platform. The conclusion drawn on such a platform may not be generalized for entire software crowd platforms. We focused TopCoder mainly for two reasons: 1) TopCoder is one of the oldest and probably the world's largest software development crowdsourcing platform with more than 1.2 million software crowd workforce and 2) TopCoders' history data is publicly available. Moreover, other software development crowdsourcing platforms do not yet provide their history data.

VIII. CONCLUSION AND FUTURE WORK

CCSD has become the emerging alternative to outsourcing software development tasks [61]. However, very little attention has been paid to what motivate competitive software developers to participate in CCSD. Most of the studies have focused micro-task crowdsourcing platforms, analyzing what motivates the crowd to participate in micro-tasks. To this end, this article presents in-depth investigation on CCSD focusing only software developers who participate and who don't want to participate in such competitions.

Our results indicate that direct economic rewards have little impact on competitive crowdsource software developers participation. Instead, learning, social contacts, and peer recognition are the major motivational drivers to participate in such competitions. Additionally, we probe the history data of TopCoder, one of the most popular platforms for crowdsourcing software development tasks, and found that Pareto principle holds for CCSD. Our results suggest that crowdsourcing software development is still in a very early stage, most of the participants do not make the essential contribution to crowdsourcing based software development, especially for complex tasks. We suggest, to attract and keep participants, crowdsourcing platforms should be more helpful for participants' career growth. Companies want to crowdsource their tasks may post simple, but time-consuming tasks, such tasks that are easy to accomplish and the quality is not so important.

Our findings suggest that crowdsourcing platforms like TopCoder should award participants with reputation and professional growth beside monetary reward. They should also focus on simple (may/may not be time-consuming) tasks that could be resolved by inexperienced developers. We also discuss some potential research directions in Section VI based on the findings reported in Section V-D.2.

ACKNOWLEDGMENTS

The authors would like to thank the associate editor and the anonymous reviewers for their insightful comments and constructive suggestions.

REFERENCES

[1] A. Dwarakanath *et al.*, "Crowd build: A methodology for enterprise software development using crowdsourcing," in *Proc. IEEE/ACM 2nd Int. Workshop CrowdSourcing Softw. Eng.*, May 2015, pp. 8–14.

[2] K.-J. Stol and B. Fitzgerald, "Researching crowdsourcing software development: Perspectives and concerns," in *Proc. 1st Int. Workshop Crowd-Sourcing Softw. Eng. (CSI-SE)*, New York, NY, USA, 2014, pp. 7–10. doi: 10.1145/2593728.2593731.

[3] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *J. Syst. Softw.*, vol. 126, pp. 57–84, Apr. 2017.

[4] M. Z. Tunio *et al.*, "Impact of personality on task selection in crowdsourcing software development: A sorting approach," *IEEE Access*, vol. 5, pp. 18287–18294, 2017.

[5] (Jan. 2018). *TopCoder: Design & Build High-Quality Software With Crowdsourcing*. [Online]. Available: http://www.topcoder.com

[6] (Jan. 2018). *Codeforces (C): The Only Programming Contests Web 2.0 Platform*. [Online]. Available: http://www.codeforces.com

[7] (Jan. 2018). *HackerEarth: Is a Global Hub of 2M+ Developers*. [Online]. Available: http://www.hackerearth.com

[8] (Jan. 2018). *CodeChef: Is a Non-Commercial Competitive Programming Community*. [Online]. Available: http://www.codechef.com

[9] (Jan. 2018). *HackerRank: Is the Standard for Assessing Developer Skills*. [Online]. Available: http://www.hackerrank.com

[10] (Jan. 2018). *Spoj: Spher Online Judge by Sphere Research Labs*. [Online]. Available: http://www.spoj.com

[11] (Jan. 2018). *Atcoder: AtCoder is a Programming Contest Site for Anyone From Beginners to Experts*. [Online]. Available: http://www.atcoder.com

[12] (Jan. 2018). *Kaggle: Competition Platform for Predictive Modeling and Analytics*. [Online]. Available: http://www.kaggle.com

[13] (Jan. 2018). *Innocentive: Is the Global Pioneer in Crowdsourced Innovation*. [Online]. Available: http://www.innocentive.com

[14] Y. Sun, N. Wang, and Z. Peng, "Working for one penny: Understanding why people would like to participate in online tasks with low payment," *Comput. Hum. Behav.*, vol. 27, no. 2, pp. 1033–1041, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747563210003742

[15] M. G. Martinez and B. Walton, "The wisdom of crowds: The potential of online communities as a tool for data analysis," *Technovation*, vol. 34, no. 4, pp. 203–214, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166497214000182

[16] A. Dubey *et al.*, "Dynamics of software development crowdsourcing," in *Proc. IEEE 11th Int. Conf. Global Softw. Eng. (ICGSE)*, Aug. 2016, pp. 49–58.

[17] Y. Fu, H. Sun, and L. Ye, "Competition-aware task routing for contest based crowdsourced software development," in *Proc. 6th Int. Workshop Softw. Mining (SoftwareMining)*, Nov. 2017, pp. 32–39.

[18] K.-J. Stol and B. Fitzgerald, "Two's company, three's a crowd: A case study of crowdsourcing software development," in *Proc. 36th Int. Conf. Softw. Eng. (ICSE)*, New York, NY, USA, 2014, pp. 187–198. doi: 10.1145/2568225.2568249.

[19] B. Fitzgerald and K.-J. Stol, "The dos and Dont's of crowdsourcing software development," in *SOFSEM: Theory and Practice of Computer Science*, G. F. Italiano, T. Margaria-Steffen, J. Pokorný, J.-J. Quisquater, and R. Wattenhofer, Eds. Berlin, Germany: Springer, 2015, pp. 58–64.

[20] A. Khanfor, Y. Yang, G. Vesonder, G. Ruhe, and D. Messinger, "Failure prediction in crowdsourced software development," in *Proc. 24th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2017, pp. 495–504.

[21] H. G. Afridi, "Empirical investigation of correlation between rewards and crowdsource-based software developers," in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. Companion (ICSE-C)*, May 2017, pp. 80–81.

[22] M. Hossain, "Users' motivation to participate in online crowdsourcing platforms," in *Proc. Int. Conf. Innov. Manage. Technol. Res.*, May 2012, pp. 310–315.

[23] D. C. Brabham, "Moving the crowd at threadless," *Inf., Commun., Soc.*, vol. 13, no. 8, pp. 1122–1145, 2010. doi: 10.1080/13691181003624090.

[24] B. W. Boehm, "Software engineering economics," in *Software Pioneers*, M. Broy and E. Denert, Eds. New York, NY, USA: Springer-Verlag, 2002, pp. 641–686. [Online]. Available: http://dl.acm.org/citation.cfm?id=944331.944370

[25] S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp, "Motivation in software engineering: A systematic literature review," *Inf. Softw. Technol.*, vol. 50, nos. 9–10, pp. 860–878, Aug. 2008. [Online]. Available: http://oro.open.ac.uk/20392/

[26] A. Dwarakanath, N. C. Shrikanth, K. Abhinav, and A. Kass, " Trustworthiness in enterprise crowdsourcing: A taxonomy & evidence from data," in *Proc. 38th Int. Conf. Softw. Eng. Companion (ICSE)*, New York, NY, USA, 2016, pp. 41–50. doi: 10.1145/2889160.2889225.

[27] N. Kaufmann, T. Schulze, and D. Veit, "More than fun and money. Worker motivation in crowdsourcing—A study on mechanical turk," in *Proc. Renaissance Inf. Technol. Sustainability Global Competitiveness, 17th Americas Conf. Inf. Syst. (AMCIS)*, Detroit, MI, USA, Aug. 2011. [Online]. Available: http://aisel.aisnet.org/amcis2011_submissions/340 and https://dblp.org/rec/bib/conf/amcis/KaufmannSV11

[28] H. Zheng, D. Li, and W. Hou, "Task design, motivation, and participation in crowdsourcing contests," *Int. J. Electron. Commerce*, vol. 15, no. 4, pp. 57–88, Jul. 2011. doi: 10.2753/JEC1086-4415150402.

[29] Y. Roth, D. C. Brabham, and J.-F. Lemoine, "Recruiting individuals to a crowdsourcing community: Applying motivational categories to an ad copy test," in *Advances in Crowdsourcing*. Cham, Switzerland: Springer, 2015, pp. 15–31.

[30] A. Namousi and D. S. Kohl, "Crowdsourcing in video games: The motivational factors of the crowd," *Comput. Games J.*, vol. 5, nos. 3–4, pp. 149–163, Dec. 2016. doi: 10.1007/s40869-016-0021-4.

[31] D. C. Brabham, "Moving the crowd at iStockphoto: The composition of the crowd and motivations for participation in a crowdsourcing application," *First Monday*, vol. 13, no. 6, pp. 1122–1145, 2008.

[32] H. Ye and A. Kankanhalli, "Solvers' participation in crowdsourcing platforms: Examining the impacts of trust, and benefit and cost factors," *J. Strategic Inf. Syst.*, vol. 26, no. 2, pp. 101–117, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0963868717300318

[33] J. Goncalves, S. Hosio, J. Rogstadius, E. Karapanos, and V. Kostakos, "Motivating participation and improving quality of contribution in ubiquitous crowdsourcing," *Comput. Netw.*, vol. 90, pp. 34–48, Oct. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128615002224

[34] Y. Sun, Y. Fang, and K. H. Lim, "Understanding sustained participation in transactional virtual communities," *Decis. Support Syst.*, vol. 53, no. 1, pp. 12–22, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923611001692

[35] R. M. Ryan and E. L. Deci, "Intrinsic and extrinsic motivations: Classic definitions and new directions," *Contemp. Educ. Psychol.*, vol. 25, no. 1, pp. 54–67, 2000.

[36] K. Lakhani, L. Bo Jeppesen, P. Lohse, and J. A. Panetta, "The value of openness in scientific problem solving," Tech. Rep., Jan. 2007.

[37] L. B. Jeppesen, K. R. Lakhani, P. A. Lohse, and J. A. Panetta, "The value of openness in scientific problem solving," Work. Paper, 2007.

[38] M. G. Martinez, "Solver engagement in knowledge sharing in crowdsourcing communities: Exploring the link to creativity," *Res. Policy*, vol. 44, no. 8, pp. 1419–1430, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0048733315000864

[39] M. G. Martinez, "Inspiring crowdsourcing communities to create novel solutions: Competition design and the mediating role of trust," *Technol. Forecasting Social Change*, vol. 117, pp. 296–304, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0040162516306849

[40] K. Lakhani and G. R. Wolf, "Why hackers do what they do: Understanding motivation and effort in free/open source software projects," *Perspect. Free Open Source Softw.*, Sep. 2003.

[41] G. Hertel, S. Niedner, and S. Herrmann, "Motivation of software developers in open source projects: An Internet-based survey of contributors to the Linux kernel," *Res. Policy*, vol. 32, no. 7, pp. 1159–1177, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0048733303000477

[42] Y. Ye and K. Kishida, "Toward an understanding of the motivation open source software developers," in *Proc. 25th Int. Conf. Softw. Eng. (ICSE)*, Washington, DC, USA, 2003, pp. 419–429. [Online]. Available: http://dl.acm.org/citation.cfm?id=776816.776867

[43] A. Hars and S. Ou, "Working for free? Motivations for participating in open-source projects," *Int. J. Electron. Commerce*, vol. 6, no. 3, pp. 25–39, Apr. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=1286960.1286963

[44] E. Capra, C. Francalanci, F. Merlo, and C. R. Lamastra, "A survey on firms' participation in open source community projects," in *Open Source Ecosystems: Diverse Communities Interacting*, C. Boldyreff, K. Crowston, B. Lundell, and A. I. Wasserman, Eds. Berlin, Germany: Springer, 2009, pp. 225–236.
[45] K. Mao, Y. Yang, Q. Wang, Y. Jia, and M. Harman, "Developer recommendation for crowdsourced software development tasks," in *Proc. IEEE Symp. Service Oriented Syst. Eng.*, Apr. 2015, pp. 347–356.
[46] T. Aitamurto, H. Landemore, and J. S. Galli, "Unmasking the crowd: Participants' motivation factors, expectations, and profile in a crowdsourced law reform," *Inf., Commun. Soc.*, vol. 20, no. 8, pp. 1239–1260, 2017. doi: 10.1080/1369118X.2016.1228993.
[47] M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemp. Phys.*, vol. 46, no. 5, pp. 323–351, Dec. 2005. doi: 10.1080/00107510500052444.
[48] N. C. Schaeffer and J. Dykema, "Questions for surveys: Current trends and future directions," *Public Opinion Quart.*, vol. 75, no. 5, pp. 909–961, 2011. [Online]. Available: http://www.jstor.org/stable/41345917
[49] S. Fernandez and R. B. Svensson, "A survey of practitioners use of open innovation," in *Proc. 43rd Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug./Sep. 2017, pp. 305–312.
[50] C. Hendry, "Research methods in health. Investigating health and health services (Second Edition) research methods in health. Investigating Health and Health Services Ann Bowling Open University Press 429 $22.50 0335206433 0335206433," *Nurse Res.*, vol. 10, pp. 81–82, Dec. 2002.
[51] K. L. Florczak, "Book review: The practice of nursing research: Conduct, critique, and utilization," *Nursing Sci.*, vol. 18, pp. 365–366, Oct. 2005.
[52] A. Mohamed and M. Kamal, "Association of student's position in a classroom and student's academic performance using ANOVA," in *Proc. 5th Int. Conf. E-Learn. (ECONF)*, Oct. 2015, pp. 392–395.
[53] M. Norris and L. Lecavalier, "Evaluating the use of exploratory factor analysis in developmental disability psychological research," *J. Autism Develop. Disorders*, vol. 40, no. 1, pp. 8–20, Jan. 2010. doi: 10.1007/s10803-009-0816-2.
[54] G. T. Lumpkin and G. G. Dess, "Clarifying the entrepreneurial orientation construct and linking it to performance," *Acad. Manage. Rev.*, vol. 21, no. 1, pp. 135–172, 1996. [Online]. Available: http://www.jstor.org/stable/258632
[55] R. Reh, M. L. Mursidi, and N. A. A. Husin, "Reliability analysis for pilot survey in integrated survey management system," in *Proc. Malaysian Conf. Softw. Eng.*, Dec. 2011, pp. 220–222.
[56] M. Iqbal and M. Rizwan, "Application of 80/20 rule in software engineering waterfall model," in *Proc. Int. Conf. Inf. Commun. Technol.*, Aug. 2009, pp. 223–228.
[57] S. N. Gaikwad, "Daemo: A self-governed crowdsourcing marketplace," in *Proc. 28th Annual Adjunct Symp. User Interface Softw. Technol. (UIST)*, New York, NY, USA, 2015. doi: 10.1145/2815585.2815739.
[58] G. Szolnoki and D. Hoffmann, "Online, face-to-face and telephone surveys—Comparing different sampling methods in wine consumer research," *Wine Econ. Policy*, vol. 2, no. 2, pp. 57–66, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212977413000331
[59] L. van Velsen, T. van der Geest, and R. Klaassen, "Testing the usability of a personalized system: Comparing the use of interviews, questionnaires and thinking-aloud," in *Proc. IEEE Int. Commun. Conf.*, Oct. 2007, pp. 1–8.
[60] T. C. Lethbridge, S. E. Sim, and J. Singer, "Studying software engineers: Data collection techniques for software field studies," *Empirical Softw. Eng.*, vol. 10, pp. 311–341, Jul. 2005.
[61] K.-J. Stol, B. Caglayan, and B. Fitzgerald, "Competition-based crowdsourcing software development: A multi-method study from a customer perspective," *IEEE Trans. Softw. Eng.*, vol. 45, no. 3, pp. 237–260, Mar. 2019.

INAM ILLAHI graduated from the University of Sargodha, Pakistan, in 2007. He received the M.S. degree from the Chalmers University of Technology, Sweden, in 2010. He is currently pursuing the Ph.D. degree from the School of Computer Science and Technology, Beijing Institute of Technology, China. He is particularly interested in software maintenance, crowdsourcing, and machine learning.

HUI LIU received the B.S. degree in control science from Shandong University, in 2001, the M.S. degree in computer science from Shanghai University, in 2004, and the Ph.D. degree in computer science from Peking University, in 2008. He was a Visiting Research Fellow with the Centre for Research on Evolution, Search and Testing (CREST), University College London, U.K. He is currently a Professor with the School of Computer Science and Technology, Beijing Institute of Technology, China. He is particularly interested in software refactoring, AI-based software engineering, and software quality. He is also interested in developing practical tools to assist software engineers. He served on the program committees and organizing committees of prestigious conferences, such as the ICSME, RE, ICSR, and COMPSAC. He is serving as an Associate Editor for the IEEE Access and *IET Software* and a Guest Editor for *Empirical Software Engineering* and the *Journal of Systems and Software*.

QASIM UMER received the B.S. degree in computer science from Punjab University, Pakistan, in 2006, and the M.S. degree in.net distributed system development and the M.S. degree in computer science from the University of Hull, U.K., in 2009 and 2012, respectively. He is currently pursuing the Ph.D. degree in computer science with the Beijing Institute of Technology, China. He is particularly interested in machine learning, data mining, and software maintenance.

SYED ANEES HAIDER ZAIDI is currently pursuing the Ph.D. with the School of Management and Economics, Beijing Institute of Technology, China. He was a Deputy Treasurer with COMSATS University Islamabad for nine years. He has published 11 research articles in the field of management and economics.