IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 14, 2017, accepted August 25, 2017, date of publication September 6, 2017, date of current version September 27, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2747660

Impact of Personality on Task Selection in Crowdsourcing Software Development: A Sorting Approach

MUHAMMAD ZAHID TUNIO[1], HAIYONG LUO[2], WANG CONG[1], ZHAO FANG[1], ABDUL REHMAN GILAL[3], AHSANULLAH ABRO[3], AND SHAO WENHUA[1]

[1]School of Software Engineering, Beijing University of Posts and Telecommunication, Beijing 100876, China
[2]Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[3]Computer Science Department, Sukkur IBA University, Sukkur Sindh 65200, Pakistan

Corresponding author: Muhammad Zahid Tunio (zahid.tunio@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61374214, in part by the National Key Research and Development Program under Grant 2016 YFB 0502004, and in part by the Open Project of the Beijing Key Laboratory of Mobile Computing and Pervasive Device.

ABSTRACT Growing demand for software has attracted the attention of the software development industry. Crowdsourced software development has provided a new method for the software industry to produce quality software based on an open-call format. Selecting an appropriate task to develop (developer-end) or evaluate (platform-end) is one of the primary problems in this type of open-call format. Receiving or assigning an improper task to an improper crowdsource (CS) developer does not only decrease the quality of the software deliverables, but also causes overburden on both the platform and the developers. To solve this problem, sorting the tasks based on the developers' human characteristics may increase task relevancy for developers, which can accelerate efficiency and lessen complexity. Thus, this paper has conducted an empirical experiment to measure the influence of personality on task selection based on the important characteristics of a task: money, time, and type. A total of 83 students from the University of Sindh voluntarily participated in four different short-duration rounds of task development using the developed CS platform. The personality types of the participants were measured based on the Myers-Briggs type indicator. In addition, a complex network technique called weighted degree centrality was applied to identify the most suitable personality for task sorting based on money or complexity attractions (i.e., time or type). Based on the results, it can be observed that personality has a significant relationship with task selection. For instance, developers with intuitive (N) and feeling (F) personality traits are primarily focused on the time duration of a project.

INDEX TERMS Crowdsource, software development, personality types, MBTI, sorting.

I. INTRODUCTION

There is an extensive demand for software in the modern world of technology. Crowdsourcing (CS) plays an important role in rapid software development. The primary theme of CS is to offer short-schedule development with parallel and micro-tasking concepts. Crowdsourced software development (CSD) uses an open-call format online to acquire a large number of workers. The open-call format of CSD involves three types of roles: 1) the requester (i.e., the one for whom the project is undertaken), 2) the platform (i.e., the service provider), and 3) the CS developer (i.e., the person performing coding and testing). This type of call format always collects a large number of self-selected tasks. In this process, a large number of developers can register and select tasks on the platform. The platform is also responsible for evaluating the submitted tasks to determine the best solution from the developers to pay the rewards. Mao *et al.* [1], [2] stated that the selection of an effective and appropriate task from the extensively large set is a hectic activity for CS developers. It is also a tiring and time-consuming assignment for platform workers to evaluate thousands of submitted tasks from the developers. For example, Fu *et al.* [3] also maintain that locating an effective task from the submitted tasks is a difficult and time-consuming undertaking for the platform workers. Based on studies by Chilton *et al.* and Aldhahari *et al.* [4], [5], receiving or assigning an improper task to an improper CS developer may not only decrease the quality of the software deliverables, but also causes overburden on both the platform and the developers. The majority of the time, workers view fewer recent tasks posted on the CS platform because hundreds of tasks are posted daily. By considering the skills and expertise level of the CS developers, unrealistic matching of CS developers and tasks would obviously have an effect on software quality.

Synchronization between the developers' expertise and the tasks, which can benefit both the platform and the CS developers, is a serious problem [6]. Based on the understanding of Machado *et al.* [7], CSD does not only encounter technical issues, but simultaneously must address human related issues. It is said that software is made for people by people [8]. The personality of CS developers is an important element that directly impacts the results or outcomes of task development in CSD [9]. Several authors from traditional software development and CSD have discussed the significant influence of personality on efficient task development [8], [10]. By considering the importance of personality, this study has conducted an empirical experiment to measure the influence of personality on task selection based on the characteristics of CSD: money, time, or type. Huggs and Cottrell [11] asserted that the success of a task is more likely to be achieved when it meets the deadlines and budgets and satisfies the quality requirements. For example, Topcoder offers a series of tasks through a challenge-based approach to attract developers within categories in which developers may be attracted based on price, time of submission, or type of task [2]. It is impossible to generalize the task ranking used by all types of developers to make their selection. Hence, sorting the tasks based on the personality type of the CS developers may increase the task relevancy for developers, accelerating the efficiency and decreasing the complexity.

II. RELATED WORK

Crowdsourcing is distributed outsourcing to an indeterminate and usually outsized crowd of people in an open-call format. It has attracted significant attention from industry and academia [12]. When employing crowdsourcing to accomplish software development tasks, CSD faces the challenges of assigning, sorting, and searching for suitable developers for specific tasks. To date, the majority of development tasks are assigned by bidding or competition. As a result, the time required to search for a suitable task based on personal preference has wasted a significant amount of human effort, although many CSD developers would not compete for the tasks [12]. The current research of crowdsourcing is focusing on several topics, for example, how to obtain benefit from crowdsourcing [13], [14], quality deliverables [15], [16], and allocating, assigning, and searching for crowd-sourced tasks. Recently, crowdsourcing software development (CSD) has received additional attention in terms of recommending tasks to developers to attract them on a large scale. For instance, Boutsis and Kalogeraki [17] presented REACT (Real Time Scheduling for crowdsourced Task) to schedule tasks for a crowd under time constraints. It collects worker profiles and dynamically assigns tasks to suitable workers. Difallah [18] used REACT and replaced "pull" with "push" for task allocation to achieve higher quality. The authors evaluated the community network of the crowd to obtain improved performance. Simpson and Roberts [19] used an information-theoretic approach to assign workers to specific tasks in crowdsourcing by adopting the Bayesian method. A few recommender frameworks similarly suggest assignments based on general qualities, e.g., strategies that make recommendations based on a model of individual client inclinations [20].

A personality-based task suggestion approach in crowdsourcing is expected to encourage matching individual interests and abilities with the correct assignments, creating potential advantages for both participants and requesters. Participants that do not have to contribute high scan expenses or agree to problematic assignments will probably maintain higher motivation.

To recommend crowdsourcing software development tasks, a content-based technique was presented by Mao *et al.* [2]. This approach used a historical record of registration and award from the CSD to automatically match the task and developer. Snow *et al.* [21] described and proposed bias correction in crowd data in the form of modeling. The authors used a gold standard data set to estimate the accuracy of the CSD workers' model, and their method is used in micro tasking. Ambati *et al.* [22] use an implicit modeling based on the skills and interests of CSD workers to recommend classification-based tasks. Yuen *et al.* [23] proposed an approach based on task matching that will encourage and motivate CSD workers to accomplish a task consistently and over the long term. This approach is focused on the recommendation of tasks that best match the workers. Sheng *et al.* [24] stated that labeling is used as a technique for task matching, but it also has limitations. Liu *et al.* [25], Whitehill *et al.* [26], and Raykar *et al.* [27] primarily used an EM algorithm to calculate the accuracy of CSD workers by using an EM algorithm and an answer matrix to relate and map the CSD workers' quality. The determination of single labeling is the focus of subsequent studies, including [28]–[30], and [31]. According to [13], by ignoring the task requirements and their relationship with CSD worker skills, these approaches may attain undesired results. Therefore, a new approach is needed to relate the soft skills to the hard skills of the CSD workers.

To avoid the risks of assigning a task to an improper crowdsourcing software development personality type, Capretz and Ahmed [32] provided a model in which they suggested that a task must be assigned to a developer based on his or her personality type. For instance, the personality of a programmer should be introvert (I), the personality of a system analyst should be extrovert (E), and a tester should have a sensing (S) and thinking (T) personality. A software designer should have an intuitive (N) and thinking (T) personality. However, due to its non-empirical nature, the effectiveness of the model is difficult to test. Hence, for CSD, we attempt to propose an approach to assign the tasks to the developers, testers, debuggers, and coders according to their personality types. This is because the developer personality type is one of the important human aspects to ensure quality in software tasks. It has also been confirmed that a technically sound individual cannot perform to a satisfactory level unless he/she is assigned to development tasks based on his/her personality type. This type of intuition raises new challenges for crowdsourcing tasks and it requires an in-depth understanding of assigning the workers appropriate tasks that match their personal characteristics [33]. Individual performance in software development has a co-relation and direct interaction with the personality of CSD workers [34]. Caprets *et al.* [10] mentioned that assigning a worker with an appropriate personality to a task in software development best suited for his/her traits increases the successful outcome of the task. Furthermore, to evaluate latent participants by assigning appropriate tasks based on their individual preferences, we believe that a new approach to maintain the self-identification process is required.

More precisely the CSD must be integrated with the developer's personality.

The individual personality types are classified based on the Myers-Briggs Type Indicator (MBTI) test, which is a combination of four dimensions, from those 4 dimensions, there are 16 possible personality combinations, as shown in Table 1. To evaluate the personality of a CSD worker, the underlying study incorporates MBTI personality type as an instrument, because it is widely used for similar personality evaluation tasks [35]–[37].

TABLE 1. 16 MBTI personality types.

ISTJ	ISFJ	INFJ	INTJ
ISTP	ISFP	INFP	INTP
ESTP	ESFP	ENFP	ENTP
ESTJ	ESFJ	ENFJ	ENTJ

III. RESEARCH METHODOLOGY

To achieve the primary objective of this study, an experimental approach was applied to a student data set. Software engineering students from the University of Sindh (UOS) in Jamshoro, Pakistan, were involved in the experiment. First, 83 final year technically sound students were chosen for the experiment. Second, the students were enrolled in the crowdsource development course. This showed that the students were excited to participate in a real time CS development environment. The experiment was conducted on a website that was developed for this research to simulate the environment of CSD. For several reasons, this study could not use existing platforms (i.e., Topcoder or Zhubajie) for the experiment. For example, none of the currently available CS development platforms store the personality type of the developers or workers and the primary objective of this study is to measure the impact of personality. Additionally, none of the vendors were willing to disclose their clients' or developers' information to us. Therefore, a customized website was developed to satisfy the study requirements.

Data collection had two phases: personal information and task information. The participants' personal information was collected during registration, e.g., demographic, academic achievements and personality types. The Myers-Briggs Type Indicator (MBTI) [10] instrument was used to collect the personality types. The task information was composed of the selected tasks undertaken by the participants and their results. Basically, the tasks were posted on the website in four different rounds. Each round offered 20 different tasks within the coding category of application design and development.

- Rounds 1 and 3 were established to determine the personality types that were attracted to prize money rather than complexity. This means that tasks with higher complexity were given a larger amount of prize money. Conversely, less complex tasks were given a smaller amount of prize money.
- Rounds 2 and 4 were established to determine the personality types that were attracted to prize money rather than deadlines. For instance, tasks with fewer days had a larger amount of prize money. Conversely, tasks with several days had a smaller amount of prize money.

The participants were to select any tasks in each round. All tasks followed the main theme of the rounds to determine the suitable personality types. The scenario was also used to explore each personality type's attraction towards the selection of task types. Furthermore, three independent requirement engineers were asked to evaluate the tasks submitted by the participants or developers. The task ranking was announced based on the aggregated results. Nevertheless, tasks that obtained more than 80% were also considered to be efficient. To extract the patterns from the collected data, several efficient techniques were used, for instance, cross tabulation and frequency techniques were applied to identify the common behaviors within the data. The choices of participants in both rounds generated a complex network between rounds and personality types. A weighted degree centrality (WDC) metric was applied to identify the most suitable node (i.e., personality type for the tasks) in both rounds. The following equation was applied in the R-project to obtain the WDC results. Additionally, the Tnet package was used in the R-project to project the network behavior.

$$C_D^{W\alpha}(\mathrm{i}) = deg_i \times \left(\frac{strength}{deg_i}\right)^{\alpha} \tag{i}$$

IV. RESULTS AND DISCUSSION

The research experiments were divided into two exercises. The first is to discover the reason for the causal occurrence of personality type. It is mentioned in the methodology section that Rounds 1 and 3 were arranged to determine the personality types that were attracted to prize money and complexity. Similarly, Rounds 2 and 4 were executed to determine the personality types that were attracted to prize money and time. These scenarios are considered because task selection is a decision making skills that is based on cognitive behavior [38]. Hence, the key interests of this study are to identify the set of personality types that are impacted by motivation factors (i.e., prize money) and also to identify which personality types prefer to be comfortable in completing the tasks (i.e., complexity or time).

First, 332 tasks were developed by 83 participants in the four rounds of the experiment. Overall, 153 tasks were developed effectively and the remaining 179 were declared to be ineffective. It is worth noting that not all of the 153 tasks were awarded prize money although the 153 developers fulfilled the requirements of the task; however, the winner and runner-ups of each round were awarded prize money. Table 2 shows the number of successful and unsuccessful tasks in each round based on the reviewers' results.

TABLE 2. Number of successful and unsuccessful submitted tasks in each round.

	Round 1	Round 2	Round 3	Round 4	Total
Successful	38	37	37	41	153
Unsuccessful	45	46	46	42	179

A. ROUND 1 AND ROUND 3

Each round had the same 83 participants with different task rotations. Initially, 16 MBTI personality types were projected to see the behavior of the class relative to prize money or complexity. Based on the descriptive appearance of the participants (see Table 3), more than half were within the four extrovert behavior personality types: 10, 11, 12, and 13. Personality type number five (ISTP) did not appear at all within the 83 participants. Moreover, to compose a WDC network, personality type number, prize money, and complexity were considered to be the network nodes. Meanwhile, the number of iterations towards the prize money or the complexity node was considered to be the weight of the network, as shown in Figure 1.

Using only network projections could not reveal the real network information. Thus, the Tnet package was used in the R-project to extract the information behind the complex network. Based on the Round 1 results, it can be observed that the introvert personality types tended more towards the complexity node rather than the prize money node. Put differently, personality types 1, 2, 4, 6, and 8 were more likely to work on less complex tasks than tasks with less prize money. More interestingly, personality types 1 through 9, 15 and 16 did not change their selection behavior in both rounds. Whereas, certain participants with personality types 10, 11, 12, 13 and 14 changed their selection from complexity to prize money. It can be inferred that extroverts prefer higher prizes and complexities than introverts.

TABLE 3. Personality types and number of participants.

Personality type No	Personality Type	Frequency
1	ISTJ	3
2	ISFJ	5
3	INFJ	4
4	INTJ	2
5	ISTP	0
6	ISFP	2
7	INFP	4
8	INTP	3
9	ESTP	6
10	ESFP	10
11	ENFP	11
12	ENTP	13
13	ESTJ	10
14	ESFJ	4
15	ENFJ	3
16	ENTJ	3

Personality types ISTJ, ISFJ, INTJ, ISFP, INTP, ESTP and ENTJ were found to be more connected to the complexity node than the prize money node in Rounds 1 and 3. However, the INFJ, INFP, ENFP, ENTP, ESFJ and ENFJ personality types were more connected to the prize money node, as shown in Table 4.

B. ROUND 2 AND ROUND 4

In this study, Rounds 2 and 4 were designed to extract the influence of time constraints on the participants' selection ability. Time constraints are equally important factors in a successful task development. In Rounds 1 and 3, the complexity factor had a similar importance as the time factor in these rounds (i.e., Rounds 2 and 4). Hence, personality types, prize money and time were considered to be the network nodes. This type of network is called a directed network in which the personality type nodes are connected to the prize money and time nodes.

Round 2 shows different task selection behavior based on personality types. For example, ISTJ, ISFJ, INTJ, and ISFP were significantly connected to the complexity node in Rounds 1 and 3; however, they were connected to prize money in Round 2. Moreover, the INFJ, INFP, ENFP, ENTP, ESFJ, and ENFJ personality types were dominated by the

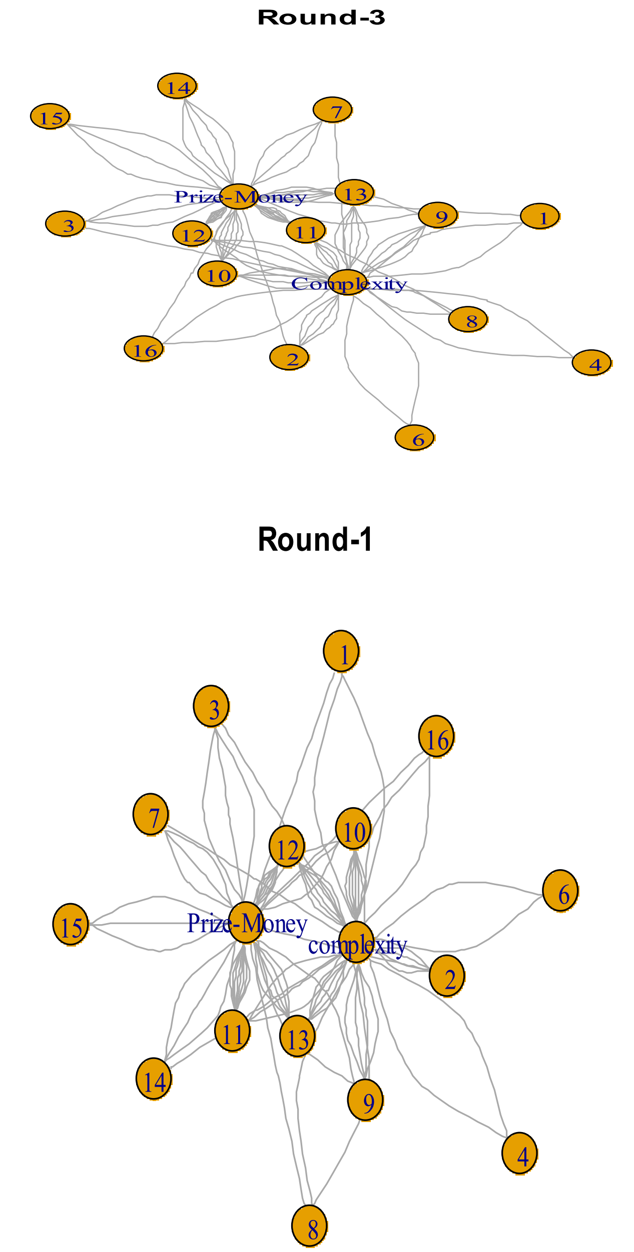


FIGURE 1. Round 1 and Round 3 network projections.

time node. By considering the results, it can be noted that intuiting (N) and feeling (F) personality traits were influencing the time node. Figures 2(a) and 2(b) depict the behavior of Round 2 based on its nodes.

However, Round 4 had a slightly different selection behavior based on the personality types, as seen after comparing Round 2 with Round 4. For instance, ISTJ, ISFP, INTP, ESFP and ENTJ were more connected to the prize money node in Round 2, whereas they were found to be dominated by the time node in Round 4. Meanwhile, personality types ISFJ, INTJ, ESTP, and ESTJ appeared more frequently in the prize money node. Similarly, the INFJ, INFP, ENFP, ENTP, ESFJ, and ENFJ personality types were more likely to work on the time nodes. Table 5 compares both rounds in detail.

TABLE 4. Round 1 and Round 3 based on personality types, prize money and complexity.

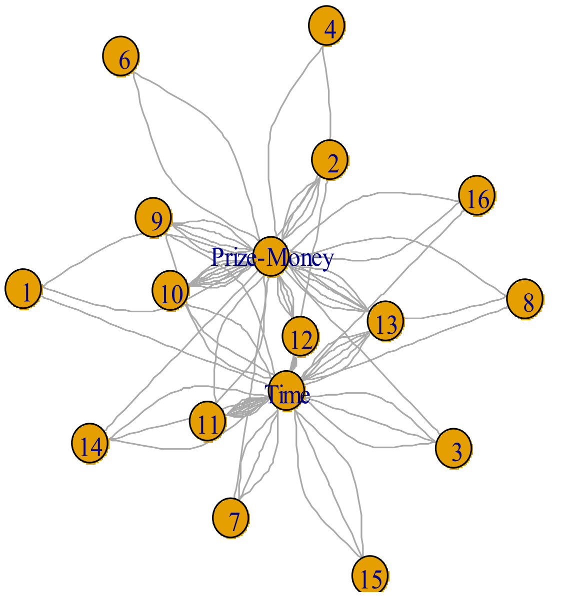
Personality type No	Personality type	Round 1		Round 3	
		Prize Money	Complexity	Prize Money	Complexity
1	ISTJ	1	2	1	2
2	ISFJ	1	4	1	4
3	INFJ	3	1	3	1
4	INTJ	0	2	0	2
5	ISTP	0	0	0	0
6	ISFP	0	2	0	2
7	INFP	3	1	3	1
8	INTP	1	2	1	2
9	ESTP	2	4	2	4
10	ESFP	3	7	6	4
11	ENFP	8	3	7	4
12	ENTP	7	6	9	4
13	ESTJ	5	5	4	6
14	ESFJ	3	1	4	0
15	ENFJ	3	0	3	0
16	ENTJ	1	2	1	2

TABLE 5. Round 2 and Round 4 based on personality types, prize money and complexity.

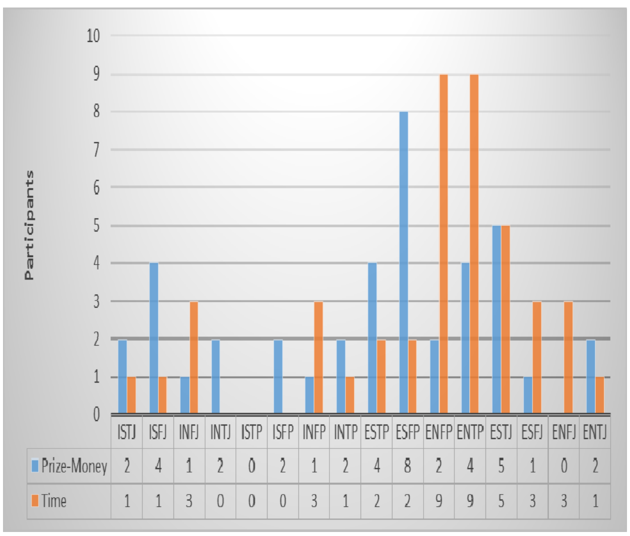
Personality type No	Personality Type	Round 2		Round 4	
		Prize Money	Time	Prize Money	Time
1	ISTJ	2	1	0	3
2	ISFJ	4	1	4	1
3	INFJ	1	3	1	3
4	INTJ	2	0	1	1
5	ISTP	0	0	0	0
6	ISFP	2	0	0	2
7	INFP	1	3	1	3
8	INTP	2	1	1	2
9	ESTP	4	2	4	2
10	ESFP	8	2	4	6
11	ENFP	2	9	3	8
12	ENTP	4	9	3	10
13	ESTJ	5	5	6	4
14	ESFJ	1	3	1	3
15	ENFJ	0	3	1	2
16	ENTJ	2	1	1	2

C. PERSONALITY TYPES, EFFECTIVE AND INEFFECTIVE OUTCOMES BASED ON PRIZE MONEY, COMPLEXITY, AND TIME NODES

The first part of the discussion presented the task selection behavior based on personality. This section focuses on the



(a)



(b)

FIGURE 2. **Round 2 network. (a) Network Projection of Round 2. (b) Prize Money and Time Nodes in Round 2.**

personality types that could produce effective outcomes based on the requirements. The results were not sufficient to finalize any personality types for task sorting or recommendation-based choices. Thus, personality type based selections were validated through task submission results (i.e., prize money, complexity and time nodes).

First, the prize money node appeared in all four rounds to compare its influence on either the complexity or time nodes. Basically, the INFJ, INFP, ENFP, ENTP, ESFJ, and ENFJ personality types appeared to be highly connected to the prize money node in Rounds 1 and 3. Based on the submission results, the ENFP personality type obtained

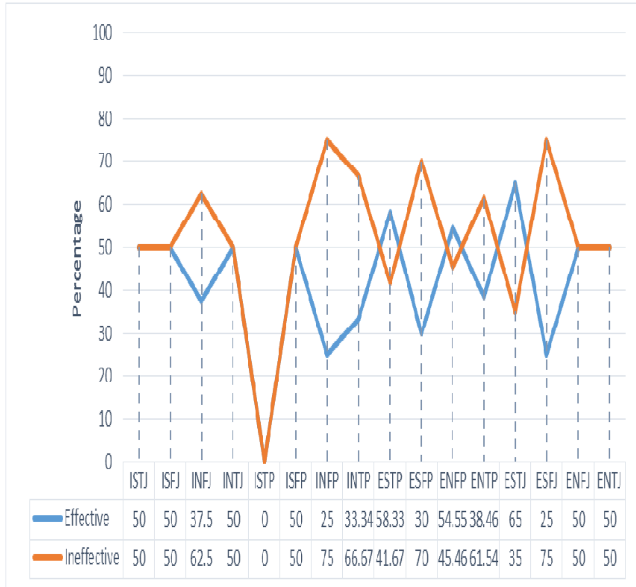


FIGURE 3. **Round 1 and Round 3 based on effective and ineffective results by personality type.**

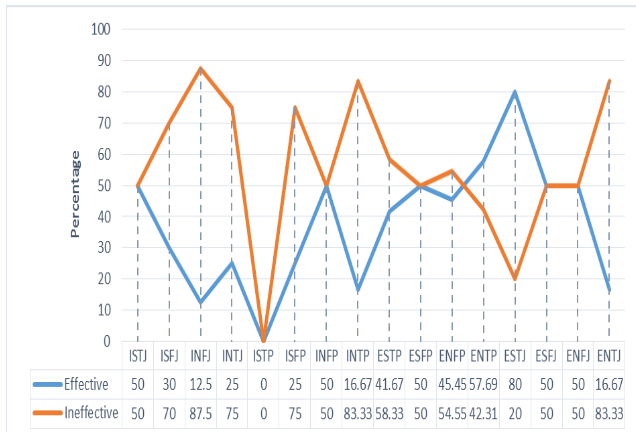


FIGURE 4. **Round 2 and Round 4 based on effective and ineffective results by personality type.**

54.55% of the effective class result. Similarly, the ENFJ personality type was found to be split between effective and ineffective classes. The remaining personality types that were connected to prize money appeared to be ineffective in obtaining the desired results. Specifically, INFJ (67% ineffective), INFP (75% ineffective), ENTP (61.5% ineffective), and ESFJ (75% ineffective) (see Figure 3 for Rounds 1 and 3). Alternatively, the ISFJ, INTJ, ESTP and ESTJ personality types were connected to the prize money node in Rounds 2 and 4. Only the ESTJ personality type was 80% effective in class tasks, while ISFJ (70% ineffective), INTJ (75% ineffective) and ESTP (58% ineffective) were ineffective in obtaining the desired results (see Figure 4 for comparative results of both rounds).

Furthermore, the ISTJ, ISFJ, INTJ, ISFP, INTP, ESTP and ENTJ personality types were connected to the complexity node in Rounds 1 and 3. Based on the reviewers' results, the ESTP personality type was found to be more effective, at 58.33%. However, ISTJ (50%), ISFJ (50%), INTJ (50%), ISFP (50%), INTP (66.67%) and ENTJ (50%) are ineffective. (see Figure 3 for more details). Similarly, the INFJ, INFP, ENFP, ENTP, ESFJ, and ENFJ personality types were connected to the time node in Rounds 2 and 4. In the same manner, the ENTP personality type was found to be more effective, while INFJ (87.5%), INFP (50%), ENFP (45.45%), ESFJ (50%) and ENFJ (50%) are ineffective. (See Figure 4 for more details).

V. CONCLUSION

This study was initiated with the goal of determining the relationship between personality and task selection. To identify the relationship, an empirical study was established with four independent rounds. Rounds 1 and 3 were arranged to extract the network of personality towards complexity or prize money. Certain personality types appeared constant for less complexity and some for a larger amount of prize money. It was also observed that the extrovert personality was more attracted to high prize money. Similarly, Rounds 2 and 4 were designed to determine the influence of personality on prize money or time. Personality types ISFJ, INTJ, ESTP, and ESTJ were more interested in selecting prize money than deadlines. Conversely, the INFJ, INFP, ENFP, ENTP, ESFJ, and ENFJ personality types were more likely to select tasks based on the timeline. Furthermore, every submitted task was not found to be effective or selected. The results section highlights that some personality types earn higher prizes than other personality types, i.e., ENFP and ENFJ. Therefore, this study concludes that personality based sorting is an effective method to reduce burdens for both the developer and the platform.

REFERENCES

[1] K. Mao, Y. Yang, Q. Wang, Y. Jia, and M. Harman, "Developer recommendation for crowdsourced software development tasks," in *Proc. 9th IEEE Int. Symp. Service-Oriented Syst. Eng. (SOSE)*, vol. 30. Mar. 2015, pp. 347–356.

[2] 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.

[3] F. Ying, H. Chen, and F. Song, "STWM: A solution to self-adaptive task-worker matching in software crowdsourcing," in *Algorithms and Architectures for Parallel Processing* (Lecture Notes in Computer Science), vol. 9528, G. Wang, A. Zomaya, G. Martinez, and K. Li, Eds. Cham, Switzerland: Springer, 2015.

[4] L. B. Chilton, J. J. Horton, R. C. Miller, and S. Azenkot, "Task search in a human computation market," in *Proc. ACM SIGKDD Workshop Hum. Comput. (HCOMP)*, New York, NY, USA, 2010, pp. 1–9.

[5] E. Aldhahri, V. Shandilya, and S. Shiva, "Towards an effective crowdsourcing recommendation system: A survey of the state-of-the-art," in *Proc. IEEE Symp. Service-Oriented Syst. Eng. (SOSE)*, Mar./Apr. 2015, pp. 372–377, doi: 10.1109/SOSE.2015.53.

[6] T. D. LaToza and A. van der Hoek, "A vision of crowd development," in *Proc. IEEE/ACM 7th IEEE Int. Conf. Softw. Eng.*, vol. 2. May 2015, pp. 563–566, doi: 10.1109/ICSE.2015.194.

[7] L. Machado, R. Prikladnicki, F. Meneguzzi, R. B. C. de Souza, and E. Carmel, "Task allocation for crowdsourcing using AI planning," in *Proc. 3rd Int. Workshop Crowdsourcing Softw. Eng. (CSISE)*, Austin, TX, USA, May 2016, pp. 36–40. [Online]. Available: http://dx.doi.org/10.1145/2897659.2897666

[8] A. R. Gilal, J. Jaafar, M. Omar, S. Basri, and A. Waqas, "A rule-based model for software development team composition: Team leader role with personality types and gender classification," *Inf. Softw. Technol.*, vol. 74, pp. 105–113, Jun. 2016.

[9] P. K. Murukannaiah, N. Ajmeri, and M. P. Singh, "Acquiring creative requirements from the crowd: Understanding the influences of personality and creative potential in Crowd RE," in *Proc. IEEE 24th Int. Requirements Eng. Conf. (RE)*, Sep. 2016, pp. 176–185.

[10] L. F. Capretz, D. Varona, and A. Raza, "Influence of personality types in software tasks choices," *Comput. Hum. Behavior*, vol. 52, pp. 373–378, Nov. 2015.

[11] B. Hughes and M. Cottrell, *Software Project Management*. New York, NY, USA: McGraw-Hill, 2002.

[12] S. Zhao, B. Shen, Y. Chen, and H. Zhong, "Towards effective developer recommendation in software crowdsourcing," in *Proc. SEKE*, 2015, pp. 326–329.

[13] A. Brew, D. Greene, and P. Cunningham, "Using crowdsourcing and active learning to track sentiment in online media," in *Proc. 19th Eur. Conf. Artif. Intell. (ECAI)*, vol. 215. Amsterdam, The Netherlands, Aug. 2010, pp. 145–150.

[14] O. F. Zaidan and C. Callison-Burch, "Crowdsourcing translation: Professional quality from non-professionals," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. 2011, pp. 1220–1229.

[15] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality control in crowdsourcing systems: Issues and directions," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 76–81, Mar. 2013.

[16] U. Hassan and E. Curry, "A capability requirements approach for predicting worker performance in crowdsourcing," in *Proc. 9th Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*, Austin, TX, USA, Oct. 2013, pp. 429–437.

[17] I. Boutsis and V. Kalogeraki, "Crowdsourcing under real-time constraints," in *Proc. IEEE 27th Int. Symp. Parallel Distrib. Process*, Boston, MA, USA, May 2013, pp. 753–764.

[18] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux, "Pick-a-crowd: Tell me what you like, and I'll tell you what to do," in *Proc. WWW*, Rio de Janeiro, Brazil, 2013, pp. 367–374.

[19] E. Simpson and S. Roberts, "Bayesian methods for intelligent task assignment in crowdsourcing systems," in *Decision Making: Uncertainty, Imperfection, Deliberation and Scalability*. Basel, Switzerland: Springer, 2015, pp. 1–32.

[20] D. Geiger and M. Schader, "Personalized task recommendation in crowdsourcing information systems—Current state of the art," *Decision Support Syst.*, vol. 65, pp. 3–16, Sep. 2014.

[21] R. Snow, B. O'Connor, and D. Jurafsky, "Cheap and fast—But is it good?: Evaluating non-expert annotations for natural language tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2008, pp. 254–263.

[22] V. Ambati, S. Vogel, and J. G. Carbonell, "Towards task recommendation in micro-task markets," in *Proc. Hum. Comput.*, 2011, pp. 80–83.

[23] M.-C. Yuen, I. King, and K.-S. Leung, "Task matching in crowdsourcing," in *Proc. 4th Int. Conf. Cyber, Phys. Social Comput. Int. Conf. Internet Things (iThings/CPSCom)*, Oct. 2011, pp. 409–412.

[24] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? Improving data quality and data mining using multiple, noisy labelers," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 614–622.

[25] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "CDAS: A crowdsourcing data analytics system," *Proc. VLDB Endowment*, vol. 5, no. 10, pp. 1040–1051, 2012.

[26] J. Whitehill, P. L. Ruvolo, T. F. Wu, J. Bergsma, and J. R. Movellan, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2035–2043.

[27] V. C. Raykar *et al.*, "Supervised learning from multiple experts: Whom to trust when everyone lies a bit," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 889–896.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.

[29] A. P. Dawidt and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Appl. Stat.*, vol. 28, no. 1, pp. 20–28, 1979.

[30] M. R. Gupta and Y. Chen, "Theory and use of the EM algorithm," *Found. Trends Signal Process.*, vol. 4, no. 3, pp. 223–296, 2010.

[31] G. J. McLachlan and T. Krishnan, "The EM algorithm and extensions," in *Wiley Series in Probability and Statistics* 2nd ed. Hoboken, NJ, USA: Wiley, 2008, doi: 10.1002/9780470191613.scard.

[32] L. F. Capretz and F. Ahmed, "Making sense of software development and personality types," *IT Prof.*, vol. 12, no. 1, pp. 6–13, 2010.

[33] G. Kazai, J. Kamps, and N. Milic-Frayling, "Worker types and personality traits in crowdsourcing relevance labels," in *Proc. CIKM*, Scotland, U.K., Oct. 2011, pp. 1941–1944.

[34] C. M. Karapıçak and O. Demirörs, "A case study on the need to consider personality types for software team formation," in *International Conference on Software Process Improvement and Capability Determination*. Berlin, Germany: Springer, 2013, pp. 120–129.

[35] S. Cruz, F. Q. B. da Silva, and L. F. Capretz, "Forty years of research on personality in software engineering: A mapping study," *Comput. Hum. Behavior*, vol. 46, pp. 94–113, May 2015.

[36] R. Valencia-Garcia, F. García-Sànchez, D. Castellanos-Nieves, J. T. Fernández-Breis, and A. Toval, "Exploitation of social semantic technology for software development team configuration," *IET Softw.*, vol. 4, no. 6, pp. 373–385, 2010.

[37] N. R. Mead, "Software engineering education: How far we've come and how far we have to go," *J. Syst. Softw.*, vol. 82, no. 4, pp. 571–575, 2009.

[38] A. R. Gilal, J. Jaffar, S. Basri, M. Omar, and M. Z. Tunio, "Making programmer suitable for team-leader: Software team composition based on personality types," in *Proc. Int. Symp. Math. Sci. Comput. Res. (iSMSC)*, May 2015, pp. 78–82.

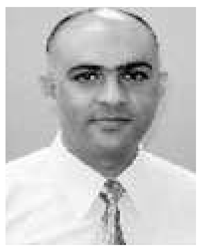
MUHAMMAD ZAHID TUNIO received the B.E. degree in computer systems engineering and the M.E. degree in information technology from the Mehran University of Engineering and Technology Jamshoro Sindh Pakistan in 2000 and 2010, respectively. He has 15 years experience in teaching at graduate and Post graduate level. He is currently a Ph.D. Scholar with the School of Software Engineering, Beijing University of Posts and Telecommunication, China. He is also an Assistant Professor with the Department of Computer Systems Engineering, Dawood University of Engineering and Technology Karachi Sindh Pakistan. His research interests are crowd source software development, pervasive devices, and mobile computing.

HAIYONG LUO received the B.S. degree from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 1989, the M.S. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunication, China, in 2002, and the Ph.D. degree in computer science from the University of Chines Academy of Sciences, Beijing, China, in 2008. He is currently an Associate Professor with the Institute of Computer Technology, Chinese Academy of Science, China. His main research interests are location-based services, pervasive computing, mobile computing, and Internet of Things.

WANG CONG received the Ph.D. degree in control theory and engineering from the University of Science Technology Beijing, China, in 2002. She was the Vice Director in charge of pattern recognition and intelligent systems with the Intelligent Science and Technology Research Center, Beijing University of Posts and Telecommunication, Beijing, from 2002 to 2011. She is currently a Professor with the School of Software Engineering, Beijing University of Post and Telecommunications. She is also the Vice Director with the Key Laboratory of Trustworthy Distributed Computing and Services, Ministry of Education. Her research interests include control theory, Internet-of-Things, and industrial systems engineering.

ZHAO FANG received the B.S. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.S. and Ph.D. degrees in computer science and technology from the Beijing University of Posts and Telecommunication, Beijing, China, in 2004 and 2009, respectively. She is currently a Professor with the School of Software Engineering, Beijing University of Posts and Telecommunication. Her research interests include mobile computing, location-based services, and computer networks.

ABDUL REHMAN GILAL received the Ph.D. degree in information technology from Universiti Teknologi Petronas, Malaysia. He is a Faculty Member with the Computer Science Department, Sukkur IBA University, Pakistan. He has been mainly researching in the field of software project management for finding the effective methods of composing software development teams. Based on his research publication track record, he has contributed in the areas of human factor in software development, complex networks, databases and data mining, and programming and cloud computing.



AHSANULLAH ABRO received the bachelor's degree in computer science from Shah Abdul Latif University, Khairpur, Pakistan, in 2006, and the M.Sc. degree in IT (by research) from Universiti Teknologi PETRONAS, Malaysia, in 2011, where he is currently pursuing the Ph.D. degree in IT. He was a Research Associate and an Assistant Professor in public institutes of higher learning and participated in various academic activities and research projects. He teaches various engineering and ICT courses to undergraduate students. His research interests include design science, development of engineering and computer science applications, and user experience evaluation of systems, products, and services.

SHAO WENHUA is currently pursuing the Ph.D. degree with the School of Software Engineering, Beijing University of Posts and Telecommunications, China. His current main interests include location-based services, pervasive computing, convolution neural networks and machine learning.