

# Personality Characteristic-based Enhanced Software Testing Levels for Crowd Outsourcing Environment

Zainab U. Kamangar<sup>1</sup>, Isma Farah Siddiqui<sup>1\*</sup>, Qasim Ali Arain<sup>1</sup>, Umair A. Kamangar<sup>1</sup>  
and Nawab Muhammad Faseeh Qureshi<sup>2\*</sup>

<sup>1</sup> Department of Software Engineering, Mehran University of Engineering and Technology,  
Jamshoro, Pakistan

[zainabumair11sw38@gmail.com, isma.farah@faculty.muett.edu.pk,  
qasim.arain@faculty.muett.edu.pk, umair11sw92@gmail.com]

<sup>2</sup> Department of Computer Education, Sungkyunkwan University, Seoul, South Korea.  
[E-mail : isma.farah@faculty.muett.edu.pk]

\*Corresponding author : Isma Farah Siddiqui & Nawab Muhammad Faseeh Qureshi

*Received April 9, 2021; revised June 15, 2021; accepted June 29, 2021;  
published August 31, 2021*

---

## Abstract

Crowd-based outsourcing is an emerging trend in testing, which integrates advantages of crowd-based outsourcing in software testing. Open call format is used to accomplish various network tasks involving different types of testing levels and techniques at various places by software testers. Crowd-sourced software testing can lead to an improper testing process as if it does not allocate the right task to the right crowd with required skills and not choose the right crowd; it can lead to huge results, which become time-consuming and challenging crowd-source manager for the identification of improper one. The primary purpose of this research is to make crowd-based outsourced software testing more effective and reliable by relating association between the software tester, personality characteristic, and different levels of software testing, i.e., unit, integration, and system, in order to find appropriate personality characteristic for required testing level. This research has shown an observed experiment to determine which software testing level suits which personality characteristic tester in a crowd-based software testing environment. A total of 1000 software testers from different software houses and firms in Pakistan were registered to perform tasks at different software testing levels. The Myers-Briggs Type Indicator (MBTI) test is used to identify each tester's personality characteristic involved in this research study.

---

**Keywords:** Crowd-based outsourcing, MBTI, software testing levels, personality characteristics, open call.

---

A preliminary version of this paper was presented at ICONI 2020, and was selected as an outstanding paper. This paper was supported by Samsung Research Fund, Sungkyunkwan University, 2018.

## 1. Introduction

Crowdsourcing is an emergent technique based on the distributed problem-solving model, which combines human and machine computation. In 2006 the term 'crowdsourcing' was combinedly devised by Howe and Robinson. It is defined as crowdsourcing as an open call format used by the organization to outsource the workers' participation on the network with undefined, online labor [1]. Mao et al. [2], [3] stated that choosing the right and efficient crowd workers from huge crowd sets is a hectic process and requires a significant amount of time to select appropriate crowd workers for required tasks evaluate their results. It is stated that the performance of the task by improper crowd worker can not only lead to low-quality deliverables but also overload the platform, further it is also stated that task may affect the quality of software if not considering relevant crowd skills and expertise for required tasks as tasks are posted in the vast amount [4], [5].

Software testing is an analyzing process to evaluate software elements' features and distinguish the alterations between current and required circumstances (i.e., defects) of software elements and improve their quality [6]. Software testing is a must for evaluating software in terms of its behavior, performance, functionality, and quality, as software testing puts away 40~50% of development efforts for reliable systems in software engineering [7].

Crowd-based outsourced software testing is the combination of software testing and crowd-based outsourcing to use software testing in the crowd-based environment (open call format) to get the advantage of recruiting globally online connected software testers. These software testers prepare the test cases for software under consideration, send them to the project manager of an organization for approval, and perform testing if their test cases get approved by the organization. However, the issue is that testing can be performed at different levels, and it is a must for a manager to know which tester performs better at which level of software testing. Software testing levels include unit, integration, and system testing level [7], [8], [9].

There is no such related research available, which can relate the software testing level with crowd-based outsourced testers for selecting suitable testers for relatable software testing synchronization [10], [11]. Therefore, to resolve this issue, experimental research is performed, which is explained in this paper, which shows that selecting suitable testers for taking part in crowd-based outsourced testing and enabling a manager to choose appropriate software tester to increase the efficiency of this process of testing task assignment. The proposed improvement in the process will lead to less time consumption and quick sorting by relating their personality types to software testing levels.

This research study's contribution can be beneficial for software companies so that they can get benefit by hiring appropriate software testers for the required testing process and can be beneficial for software testing team leader for allocating tasks to a suitable tester.

The remaining sections of this paper are organized as follows. Section 2 gives an overview of related work. Section 3 discusses the proposed methodology. Section 4 presents the solution, results, and discussion along with the experimental evaluation of the proposed system in the real indoor environment. Finally, the conclusion is presented in Section 5.

## 2. Related work

The relationship between crowd-sourced software testing techniques and personality type has been proposed [12]. It has been identified that personality type with extrovert characteristic software testers is suitable at performing tasks related to black-box testing techniques. In contrast, personality type with introvert characteristic software testers are suitable for performing white-box testing techniques. Also, it has been identified that software testers with personality types having dimension J are more suitable for performing tasks related to black-box testing techniques [12].

The method for the task assignment in a crowdsourcing environment for collaborative development has been proposed. It focuses on three factors to achieve worker's time, capacity, and task module complexity [13]. Performance can be affected by the tester's ability and the difficulty of testing tasks in crowd-based outsourcing. It is also found that a functional testing task is executed improved when executed by a skilled crowd tester compared to one with less or no experience in a crowd-based outsourcing environment; this difference between expert and beginner is more evident when the task is more complex [14].

Crowd-based outsourcing faces the daunting task of allocating different types of tasks to proper developers. In this process, there is a need to identify developers' proficiencies and assign tasks based upon those proficiencies, but a significant portion of tasks are allocated based on competition or bidding, resulting in a waste of time and human effort [15].

The current study of crowd-based outsourcing has been concentrating upon diverse areas, like; how crowd-based outsourcing is advantageous [16], [17], how to deal, assign and solve tasks of crowd-based outsourcing and quality deliverables [18], [19]. The task is based upon real-time scheduling, which uses worker profiles to allocate responsibilities, such as Real-Time Scheduling for crowdsourced tasks (REACT) [20]. Tasks through an information-theoretic approach are allocated to staff, but these methods are traditional probabilistic methods and do not establish a concrete criterion for testing tasks [21]. The individual-client inclination Model is also proposed in which some strategies are also presented to make recommendations [22]. A content-based technique was presented to ratify tasks of crowdsourcing development [3].

Implicit modeling was used to ratify crowd workers' classification-based tasks based on their interests and skills [23]. An approach is presented that focuses on recommending tasks that suit the crowd workers [24]. Sheng et al. [25] stated the limitations to labeling for task matching. Undesirable results could be produced due to workers' skills because of neglecting the relationship between tasks and their requirements [26].

According to the proposed model, the tasks are allocated to crowd workers related to their personality types [27]. A task cannot be executed if it is not allocated to a desirable personality type worker. This statement also raises new challenges for crowdsourcing tasks and involves a detailed consideration of allocating the workers with practical tasks related to their personality type [28].

It is also mentioned that the probability of a task becoming successful increases if a task is assigned to the appropriate personality type [29]. Task Assignment Model based on crowd-based outsourcing for software development has also been proposed [30]. The association between personality characteristics and task selection is identified, and it also establishes the association of personality types with both prize money and deadlines; some personality characteristics were more attracted in choosing prize money than time limit while some were

more interested in choosing tasks related to the timeline [31]. The individual personality type, which is a combination of four dimensions, as shown in **Table 1**, is classified based on the MBTI (Myers-Briggs Type Indicator) test, so there are 16 possible personality combinations shown in **Table 2**. To evaluate the personality characteristic of a crowd-based outsourced software development worker MBTI test of personality characteristics is used as an instrument, as it identifies the similarities between tasks according to personality characteristics [32], [33], [34], [35], [36], [37].

**Table 1.** MBTI Personality Characteristic's Four Dimensions

<b>Extroversion (E)</b>	<b>Introversion (I)</b>
<b>Sensing (S)</b>	<b>Intuition (N)</b>
<b>Feeling (F)</b>	<b>Thinking (T)</b>
<b>Perceiving (P)</b>	<b>Judging (J)</b>

**Table 2.** MBTI 16 Personality Characteristics

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

This research paper aims to determine which level of software testing (unit, integration, and system-level) is suitable for which personality type tester in a crowdsourced software testing environment.

### 3. Personality Characteristic-based Enhanced Software Testing Levels

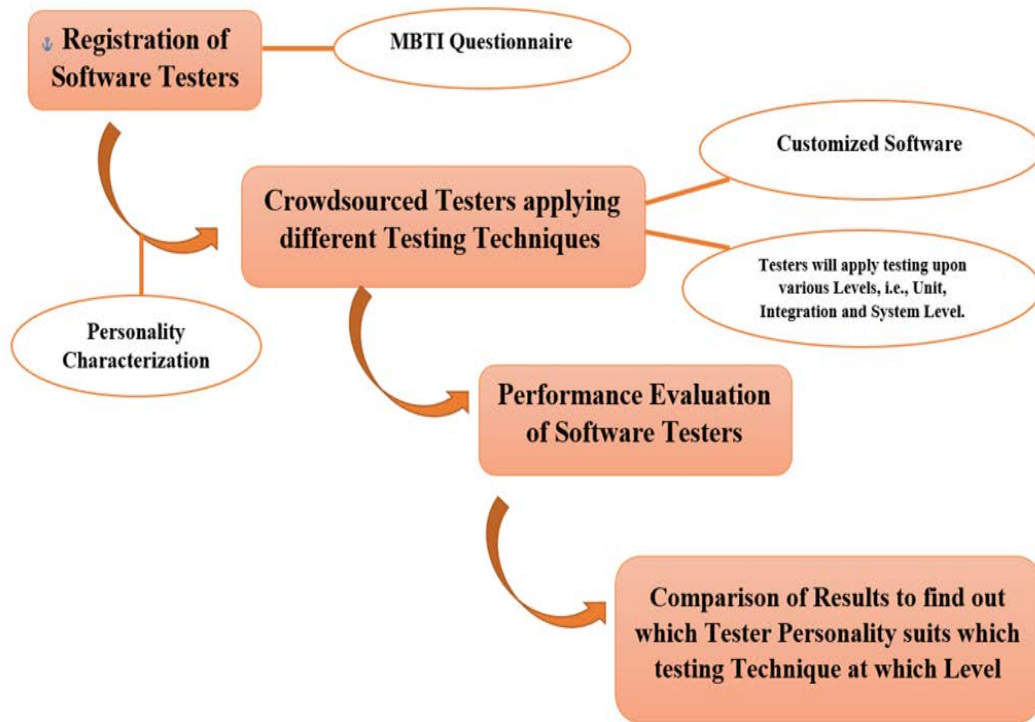
In order to fulfill research goals, a tentative tactic was integrated into the data set of testers. Software testers from various firms and software houses in Pakistan were involved in this research. Firstly, 1000 testers were registered for this research. Testers were required to fill the registration form, which contains a questionnaire based on their personality characteristic. The distinct personality characteristics are classified based on the MBTI (Myers-Briggs Type Indicator) test [29]. In this study, a crowd tester's personality is evaluated by integrating MBTI as an instrument to identify personality characteristic, as it is extensively used for similar assessment tasks based on personality characteristics.

The study was conducted on a customized software program by providing testers with its code and tasks. Testers were asked to complete those tasks on provided software to accomplish unit, integration, and system-level software testing aspects. Customized software was also developed to set the tasks to accomplish the tasks related to levels to satisfy the study requirements. Tasks were given in three different rounds to satisfy requirements concerning each level of software testing. Each tester has to do every task on the given software program.

- Round 1 contains five tasks of each unit and integration testing level to compare the average score of each personality characteristic with unit and integration testing level.
- Round 2 contains five tasks of each integration and system testing level to compare the average score of each personality characteristic with integration and system testing

level.

- Round 3 contains five tasks of each unit and system testing level to compare the average score of each personality characteristic with unit and system testing level.



**Fig. 1.** Research Methodology

The testers were asked to solve each and every task in each round, and then the score was given based on the number of bugs found. After that, the average score was calculated for each personality characteristic. A comparison was made on their average score by calculating their percentage and observed which distinct personality characteristic suits at which software testing level.

The equation used to calculate the average score of personality characteristic is given in equation (1), where APC stands for an average score of personality characteristic and PC stands for personality characteristic. Further, the percentage of an average score of personality characteristic is calculated by equation (2).

$$APC = \frac{\sum(\text{Score of PC})}{\text{frequency of PC}} \quad (1)$$

$$APC \% = \frac{APC}{\text{Total score}} \times 100 \quad (2)$$

## 4. Results and Discussion

In this study, to accomplish the research experiment, the study was conducted on a customized software program and set up defined tasks of each level of testing in different rounds. Firstly, 1000 testers were registered from different firms and software houses in Pakistan. While registering the testers, they were required to fill the questionnaire and the registration form by which their personality characteristic was identified using the MBTI test indicator. **Table 3** shows the personality characteristic and the consecutive number of testers registered in that personality characteristic. The study was conducted by providing testers with a customized software program code, and tasks. Customized software was also developed to set the tasks to accomplish the tasks related to levels in order to satisfy the study requirements (unit, integration, and system levels of software testing aspects).

**Table 3.** Personality characteristic and number of testers registered having corresponding personality characteristic

Personality Characteristic No:	Personality Characteristic	Frequency
1	ISTP	50
2	ISFP	27
3	INFP	88
4	INTP	65
5	ISTJ	65
6	ISFJ	102
7	INFJ	25
8	INTJ	56
9	ESTP	38
10	ESFP	49
11	ENFP	88
12	ENTP	99
13	ESTJ	54
14	ESFJ	100
15	ENFJ	55
16	ENTJ	39

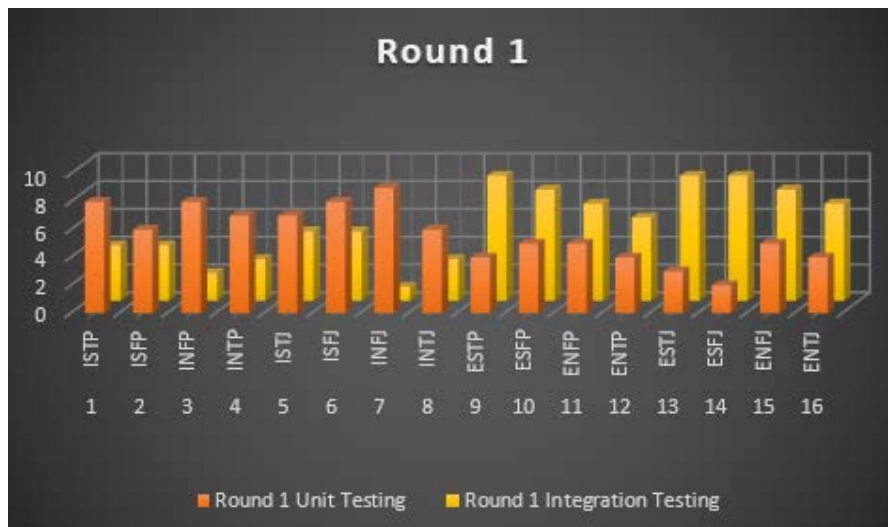
Tasks were given in three different rounds to satisfy requirements concerning each level of software testing. Each tester has to do every task on the given software program.

### 4.1 Round 1

Round 1 contains five tasks of each unit and integration testing level to compare the average score of each personality characteristic with unit and integration testing level in order to find which personality characteristic software tester performs which software testing level better. **Table 4** indicates Round 1, which contains tasks related to unit and integration testing level based on a scale from 1 to 10.

**Table 4.** Personality characteristic and consecutive score average of testers in Round 1 (Tasks of unit and integration testing level) based on scale (1-10)

Personality Characteristic No:	Personality Characteristic	Software Testing Levels (Round 1)	
		Unit Testing	Integration Testing
1	ISTP	8	4
2	ISFP	6	4
3	INFP	8	2
4	INTP	7	3
5	ISTJ	7	5
6	ISFJ	8	5
7	INFJ	9	1
8	INTJ	6	3
9	ESTP	4	9
10	ESFP	5	8
11	ENFP	5	7
12	ENTP	4	6
13	ESTJ	3	9
14	ESFJ	2	9
15	ENFJ	5	8
16	ENTJ	4	7

**Fig. 2.** Round 1 (Unit and Integration Testing Levels) Results by Personality Characteristics

From **Fig. 2**, it is shown that in Round 1, tasks related to unit and integration testing levels are carried out, and a score was given based on scale ranges from 1 to 10. After that average score was calculated for the particular personality characteristic, and it is identified that testers having introvert personality characteristics are better at executing unit testing level as compared to testers having extrovert personality characteristics. In contrast, testers who have extrovert personality characteristics are better at executing integration testing levels than testers who have introvert personality characteristics.

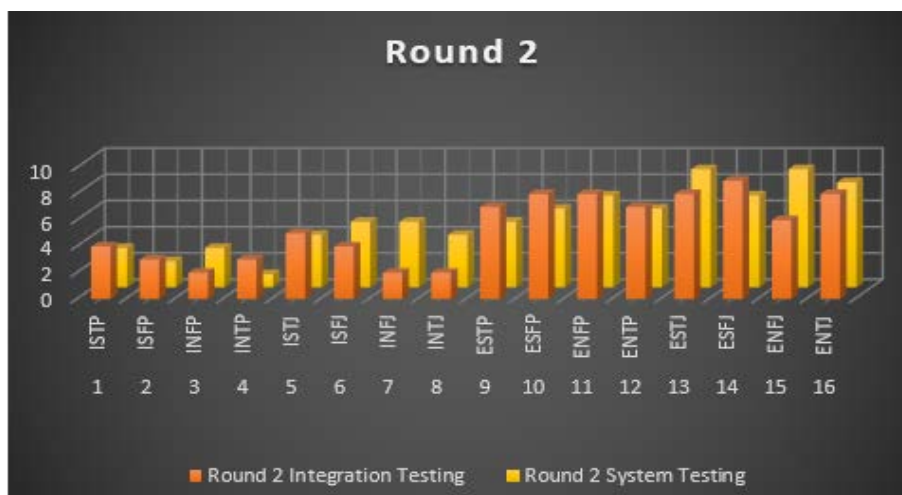


### 4.2 Round 2

Round 2 contains five tasks of each integration and system testing level to compare the average score of each personality characteristic with integration and system testing level in order to find which personality characteristic software tester performs which software testing level better. **Table 5** indicates Round 2, which contains tasks related to integration and system testing based on a scale from 1 to 10.

**Table 5.** Personality characteristic and consecutive score average of testers in Round 2 (Tasks of integration and system testing level) based on scale (1-10)

Personality Characteristic No:	Personality Characteristic	Software Testing Levels (Round 2)	
		Integration Testing	System Testing
1	ISTP	4	3
2	ISFP	3	2
3	INFP	2	3
4	INTP	3	1
5	ISTJ	5	4
6	ISFJ	4	5
7	INFJ	2	5
8	INTJ	2	4
9	ESTP	7	5
10	ESFP	8	6
11	ENFP	8	7
12	ENTP	7	6
13	ESTJ	8	9
14	ESFJ	9	7
15	ENFJ	6	9
16	ENTJ	8	8



**Fig. 3.** Round 2 (Integration and System Testing Levels) Results by Personality Characteristics

From **Fig. 3**, it is shown that in Round 2, tasks related to integration and system testing levels are carried out, and a score was given based on scale ranges from 1 to 10. After that



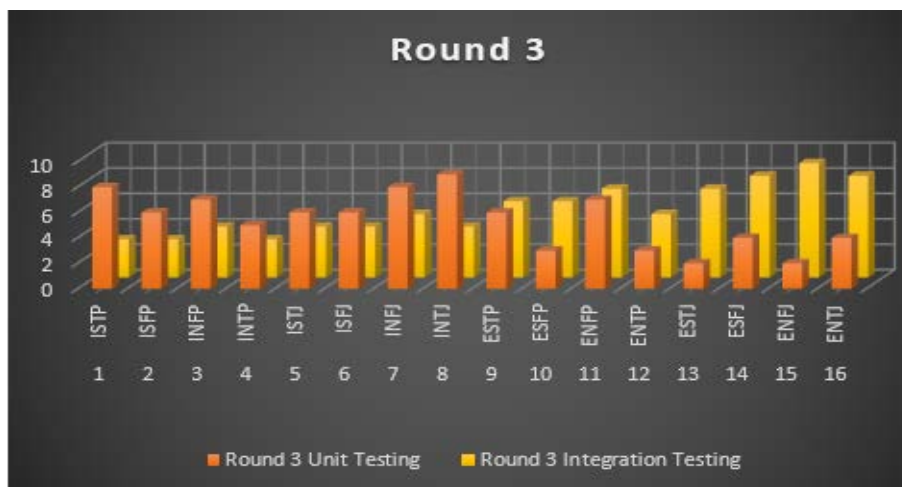
average score was calculated for the particular personality characteristic, and it is identified that testers having extrovert personality characteristics are better at executing integration and system testing level as compared to testers having introvert personality characteristics.

### 4.3 Round 3

Round 3 contains five tasks of each unit and system testing level to compare the average score of each personality characteristic with the unit and system testing level to find which personality characteristic suits at which software testing level better. **Table 6** indicates Round 3, which contains tasks related to unit and system testing levels based on a scale from 1 to 10.

**Table 6.** Personality characteristic and consecutive score average of testers in Round 3 (Tasks of unit and system testing level) based on scale (1-10)

Personality Characteristic No:	Personality Characteristic	Software Testing Levels (Round 3)	
		Unit Testing	System Testing
1	ISTP	8	3
2	ISFP	6	3
3	INFP	7	4
4	INTP	5	3
5	ISTJ	6	4
6	ISFJ	6	4
7	INFJ	8	5
8	INTJ	9	4
9	ESTP	6	6
10	ESFP	3	6
11	ENFP	7	7
12	ENTP	3	5
13	ESTJ	2	7
14	ESFJ	4	8
15	ENFJ	2	9
16	ENTJ	4	8



**Fig. 4.** Round 3 (Unit and System Testing Levels) Results by Personality Characteristics

From **Fig. 4**, it is shown that in Round 3, tasks related to unit and system testing levels are carried out, and a score was given based on scale ranges from 1 to 10. After that average score was calculated for the particular personality characteristic, and it is identified that testers having introvert personality characteristics are better at executing unit testing level as compared to testers having extrovert personality characteristics. In contrast, testers who have extrovert personality characteristics are better at executing integration testing levels than testers who have introvert personality characteristics.

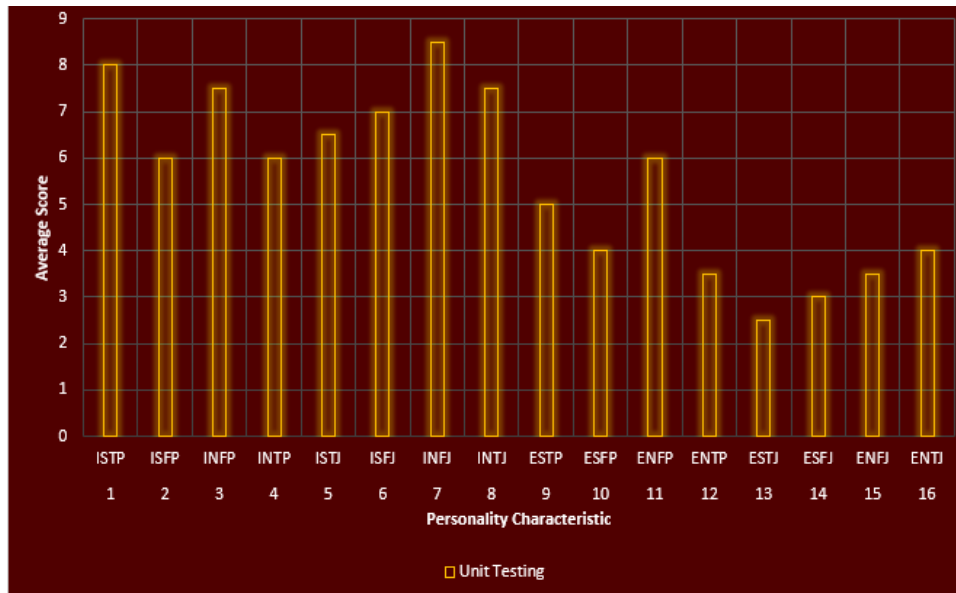
#### 4.4 Comparison of personality characteristic with respect to software testing levels based on average score

This section focuses on comparing personality characteristics and software testing level based on average score from rounds 1, 2, and 3 to find which personality characteristic tester performs tasks related to which software testing level better. **Table 7** indicates the average score of Unit, Integration, and System testing level contained in Round 1, 2, and 3 to find which personality characteristic executes tasks of which software testing level better based on a scale from 1 to 10.

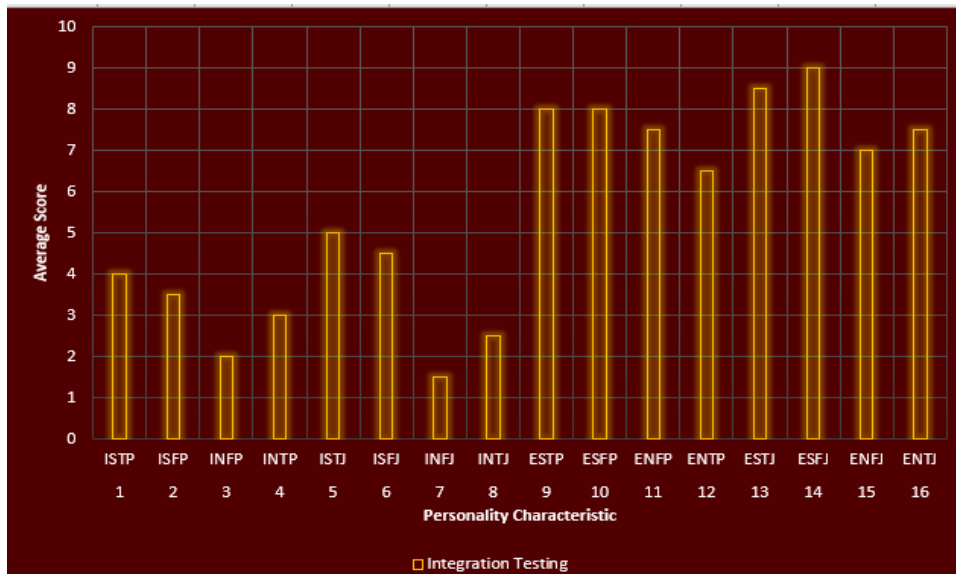
From **Table 7**, it is concluded that testers with introvert personality characteristics are more appropriate for unit testing level than extrovert personality characteristics except for ENFP. In contrast, the testers who have extrovert personality characteristics are better at executing integration and system testing levels. Further it is also analyzed that testers with personality characteristics having S (Sensing) dimension are better at executing tasks related to Integration testing level, also it is analyzed that testers with personality characteristics having J (Judging) dimension are better at executing tasks related to System testing level.

**Table 7.** Personality characteristic and consecutive score average of testers in different testing levels based on scale (1-10)

Personality Characteristic No:	Personality Characteristic	Software Testing Levels (Round 3)		
		Unit Testing	Integration Testing	System Testing
1	ISTP	8	4	3
2	ISFP	6	3.5	2.5
3	INFP	7.5	2	3.5
4	INTP	6	3	2
5	ISTJ	6.5	5	4
6	ISFJ	7	4.5	4.5
7	INFJ	8.5	1.5	5
8	INTJ	7.5	2.5	4
9	ESTP	5	8	5.5
10	ESFP	4	8	6
11	ENFP	6	7.5	7
12	ENTP	3.5	6.5	5.5
13	ESTJ	2.5	8.5	8
14	ESFJ	3	9	7.5
15	ENFJ	3.5	7	9
16	ENTJ	4	7.5	8

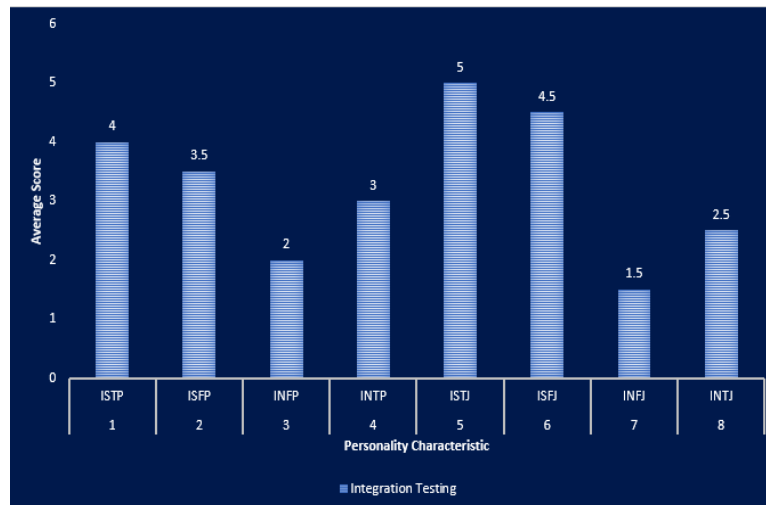


**Fig. 5.** Unit testing level Results by Personality Characteristics



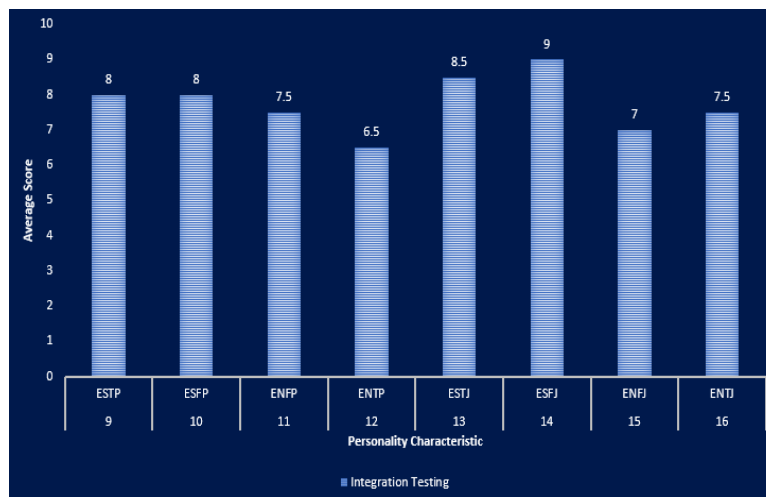
**Fig. 6.** Integration testing level Results by Personality Characteristics

From **Fig. 5**, it is shown that testers who have introvert personality characteristics are appropriate for unit testing level as compared to extrovert personality characteristics except for ENFP. From **Fig. 6**, it is shown that the testers who have extrovert personality characteristics are better at executing tasks related to integration testing level.



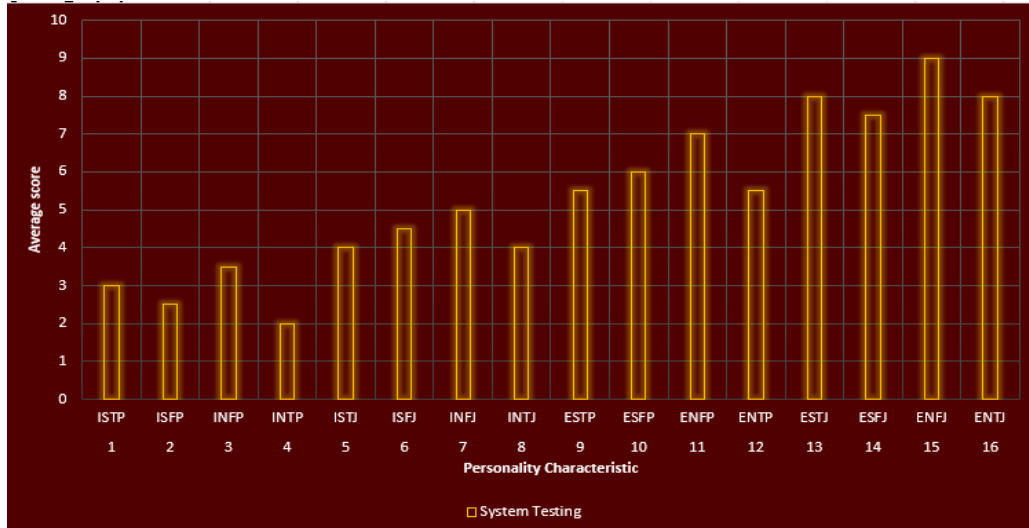
**Fig. 7.** Integration testing level Results by Introvert Personality Characteristics

From **Fig. 7**, it is shown that introvert personality characteristics having S (sensing) dimension executes Integration Testing Level improved than introvert personality characteristics which do not have S (sensing) dimension, means ISTP, ISFP, ISTJ and ISFJ personality characteristic testers are suitable in executing integration testing level than INFP, INTP, INFJ and INTJ personality characteristic testers.



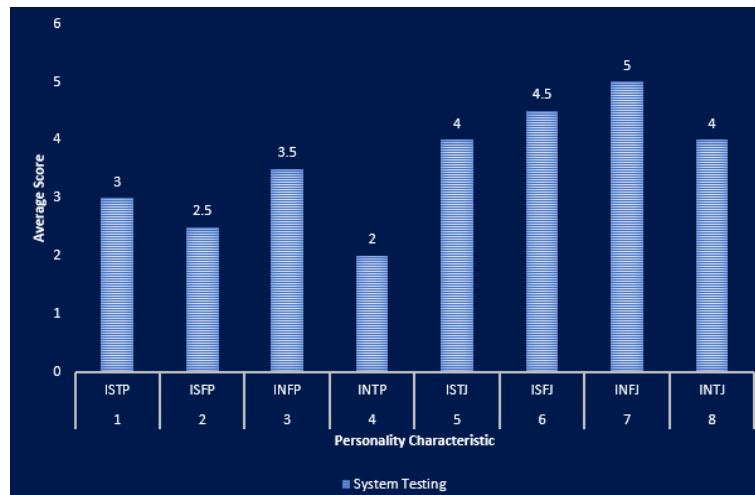
**Fig. 8.** Integration testing level Results by Extrovert Personality Characteristics

From **Fig. 8**, it is shown that extrovert personality characteristics having S (sensing) dimension executes Integration Testing Level tasks improved than extrovert personality characteristics which do not have S (sensing) dimension, means ESTP, ESFP, ESTJ and ESFJ are suitable in executing integration testing level than ENFP, ENTP, ENFJ and ENTJ personality characteristic testers.



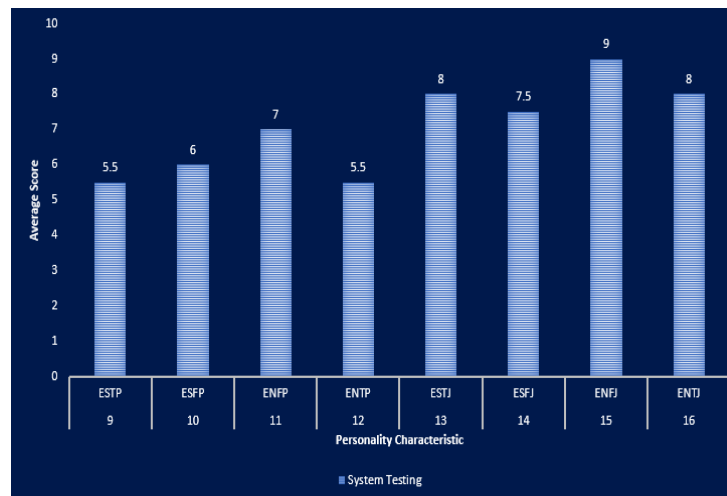
**Fig. 9.** System testing level Results by Personality Characteristics

From **Fig. 9**, it is shown that the testers, who have extrovert personality characteristics, are better at executing tasks related to system testing level than introvert personality characteristics.



**Fig. 10.** System testing level Results by Introvert Personality Characteristics

From **Fig. 10**, it is shown that introvert personality characteristics having J (judging) dimension executes System testing level improved than introvert personality characteristics which do not have J (judging) dimension, means INFJ, ISFJ, ISTJ, and INTJ personality characteristic testers are suitable in executing system testing level than INFP, ISFP, ISTP and INTP personality characteristic testers.



**Fig. 11.** System testing level Results by Extrovert Personality Characteristics

From **Fig. 11**, it is shown that extrovert personality characteristics having J (judging) dimension executes System Testing Level tasks improved than extrovert personality characteristics which do not have J (judging) dimension, means ESFJ, ENFJ, ESTJ and ENTJ are suitable in executing system testing level than ENFP, ESFP, ESTP and ENTP personality characteristic testers.

**Table 8**, indicates the percentage of average score of Unit, Integration, and System testing levels to compare overall impact to find which personality characteristic tester executes tasks of which software testing level better.

**Table 8.** Personality Characteristic and percentage of consecutive score average of testers

Personality Characteristic No:	Personality Characteristic	Software Testing Levels (Round 3)		
		Unit Testing	Integration Testing	System Testing
1	ISTP	80	40	30
2	ISFP	60	35	25
3	INFP	75	20	35
4	INTP	60	30	20
5	ISTJ	65	50	40
6	ISFJ	70	45	45
7	INFJ	85	15	50
8	INTJ	75	25	40
9	ESTP	50	80	55
10	ESFP	40	80	60
11	ENFP	60	75	70
12	ENTP	35	65	55
13	ESTJ	25	85	80
14	ESFJ	30	90	75
15	ENFJ	35	70	90
16	ENTJ	40	75	80



**Fig. 12.** Unit, Integration and System testing level Results by Personality Characteristics

From **Fig. 12**, it is concluded that testers who have introvert personality characteristic i.e., ISTP (80%), ISFP (60%), INFP (75%), INTP (60%), ISTJ (65%), ISFJ (70%), INFJ (85%) or INTJ (75%) are more appropriate for unit testing level as compared to extrovert personality characteristic except ENFP (60%). While the testers, who have extrovert personality characteristic i.e., ESTP (80%), ESFP (80%), ENFP(75%), ENTP (65%), ESTJ (85%), ESFJ (90%), ENFJ (70%) or ENTJ (75%) are better in performing tasks related to integration testing level as compared to introvert personality characteristic, similarly, the testers, who have extrovert personality characteristic i.e., ESTP (55%), ESFP (60%), ENFP(70%), ENTP (55%), ESTJ (80%), ESFJ (75%), ENFJ (90%) or ENTJ (80%) are better in performing tasks related to system testing level as compared to introvert personality characteristic. Further it is also analyzed that testers with personality characteristics having S (Sensing) dimension are better at executing tasks related to Integration testing level, means ESTP, ESFP, ESTJ and ESFJ personality characteristic testers are suitable in executing integration testing level than ENFP, ENTP, ENFJ and ENTJ personality characteristic testers, similarly testers with personality characteristic ISTP, ISFP, ISTJ and ISFJ are suitable in executing integration testing level than INFP, INTP, INFJ and INTJ personality characteristic testers, also it is analyzed that testers with personality characteristics having J (Judging) dimension are better at executing tasks related to System testing level, means ENFJ, ESFJ, ESTJ, and ENTJ personality characteristic testers are suitable in executing system testing level than ENFP, ESFP, ESTP and ENTP personality characteristic testers, similarly testers with personality characteristic ISFJ, INFJ, ISTJ and INTJ are suitable in executing system testing level than INFP, ISFP, ISTP and INTP personality characteristic testers.

#### 4.5 Limitations of this Study

This study has potential limitations as dataset is limited and by increasing its frequency can lead to more appropriate results and more suitability points. Tester's experience on particular testing level can also affects its suitability comparison that which personality characteristic tester performs better at which software testing level.



## 5. Conclusion

This study's primary objective was to govern the association among software testing levels and tester with different personality characteristics, which will show which personality characteristic testers are appropriate for software testing level. Software testing level-related tasks should be allocated to related personality characteristic tester. From this research study it is concluded that testers who have introvert personality characteristic i.e., ISTP, ISFP, INFP, INTP, ISTJ, ISFJ, INFJ or INTJ are more appropriate for unit testing level as compared to extrovert personality characteristic except ENFP i.e., ESTP, ESFP, ENTP, ESTJ, ESFJ, ENFJ or ENTJ. While the testers, who have extrovert personality characteristic i.e., ESTP, ESFP, ENFP, ENTP, ESTJ, ESFJ, ENFJ or ENTJ are better in performing tasks related to integration testing level as compared to introvert personality characteristic, similarly, the testers, who have extrovert personality characteristic i.e., ESTP, ESFP, ENFP, ENTP, ESTJ, ESFJ, ENFJ or ENTJ are better in performing tasks related to system testing level as compared to introvert personality characteristic. Further it is also analyzed that testers with personality characteristics having S (Sensing) dimension are better at executing tasks related to Integration testing level, means ESTP, ESFP, ESTJ and ESFJ personality characteristic testers are suitable in executing integration testing level than ENFP, ENTP, ENFJ and ENTJ personality characteristic testers, similarly testers with personality characteristic ISTP, ISFP, ISTJ and ISFJ are suitable in executing integration testing level than INFP, INTP, INFJ and INTJ personality characteristic testers, also it is analyzed that testers with personality characteristics having J (Judging) dimension are better at executing tasks related to System testing level, means ENFJ, ESFJ, ESTJ, and ENTJ personality characteristic testers are suitable in executing system testing level than ENFP, ESFP, ESTP and ENTP personality characteristic testers, similarly testers with personality characteristic ISFJ, INFJ, ISTJ and INTJ are suitable in executing system testing level than INFP, ISFP, ISTP and INTP personality characteristic testers.

## References

- [1] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006. [Article \(CrossRefLink\)](#)
- [2] K. Mao, Y. Yang, Q. Wang, Y. Jia, and M. Harman, "Developer recommendation for crowdsourced software development tasks," in *Proc. of 9<sup>th</sup> IEEE Int. Symp. Service-oriented Syst. Eng. (SOSE)*, vol. 30, pp. 347-356, Mar. 2015.
- [3] Ke Mao, Licia Capra, Mark Harman, and Yue Jia, "A Survey of the Use of Crowdsourcing in Software Engineering," *Journal of Systems and Software*, vol. 126, pp. 57-84, Apr. 2017. [Article \(CrossRefLink\)](#)
- [4] L. B. Chilton, J. J. Horton, R. C. Miller, and S. Azenkot, "Task search in a human computation market," in *Proc. of the ACM SIGKDD Workshop on Human Computation*, Washington, DC, pp. 1-9, 2010. [Article \(CrossRefLink\)](#)
- [5] E. Aldhahri, V. Shandilya, and S. Shiva, "Towards an effective crowdsourcing recommendation system: a survey of the state-of-the-art," in *Proc. of 2015 IEEE Symposium on Service-Oriented System Engineering, San Francisco Bay, CA*, pp. 372-377, 2015. [Article \(CrossRefLink\)](#)
- [6] IEEE Guide for Software Verification and Validation Plans, IEEE Std 1059-1993, pp.1-87, 28 April 1994. [Article \(CrossRefLink\)](#)
- [7] Luo, Lu, "Software testing techniques: technology maturation and research strategy," *Class report for*, 2001. [Article \(CrossRefLink\)](#)
- [8] K. Sneha and G. M. Malle, "Research on software testing techniques and software automation testing tools," in *Proc. of Int. Conf. Energy, Commun. Data Anal. Soft Comput. (ICECDS) 2017*,

- pp. 77–81, 2017. [Article \(CrossRefLink\)](#)
- [9] C. Padmini, “Beginners Guide to Software Testing,” pp. 1–41, 2013. [Article \(CrossRefLink\)](#)
- [10] T. D. LaToza and A. van der Hoek, “A vision of crowd development,” in *Proc. of IEEE/ACM 7th IEEE Int. Conf. Softw. Eng.*, vol. 2, pp. 563–566, May 2015. [Article \(CrossRefLink\)](#)
- [11] L. Machado, R. Prikladnicki, F. Meneguzzi, R. B. C. de Souza, and E. Carmel, “Task allocation for crowdsourcing using AI planning,” in *Proc. of 3rd Int. Workshop Crowdsourcing Softw. Eng. (CSISE)*, Austin, TX, USA, pp. 36-40, May 2016. [Article \(CrossRefLink\)](#)
- [12] Zainab Umair Kamangar, Umair Ayaz Kamangar, Qasim Ali, Isma Farah, Shahzad Nizamani and Tauha Husain Ali, “To enhance Effectiveness of Crowdsourced Software Testing by applying Personality Types,” in *Proc. of ICSIE '19 8th International Conference on Software and Information Engineering*, Cairo, Egypt, pp. 15-19, April 2019. [Article \(CrossRefLink\)](#)
- [13] Yu, Dunhui & Zhou, Zhuang & Wang, Yi, “Crowdsourcing Software Task Assignment Method for Collaborative Development,” *IEEE Access*, vol. 7, pp. 35743-35754, March 2019. [Article \(CrossRefLink\)](#)
- [14] Leicht, N., Rhyn M., Hansbauer G., “Can laymen outperform experts? The effects of user expertise and task design in crowdsourced Software Testing,” in *Proc. of Twenty-Fourth European Conference on Information Systems (ECIS 2016)*, İstanbul, Turkey, pp. 8, 2016. [Article \(CrossRefLink\)](#)
- [15] S. Zhao, B. Shen, Y. Chen, and H. Zhong, “Towards effective developer recommendation in software crowdsourcing,” *Proc. SEKE*, pp. 326–329, 2015. [Article \(CrossRefLink\)](#)
- [16] A. Brew, D. Greene, and P. Cunningham, “Using crowdsourcing and active learning to track sentiment in online media,” in *Proc. of 19th Eur. Conf. Artif. Intell. (ECAI)*, Amsterdam, The Netherlands, vol. 215, pp. 145–150, Aug. 2010. [Article \(CrossRefLink\)](#)
- [17] O. F. Zaidan and C. Callison-Burch, “Crowdsourcing translation: Professional quality from non-professionals,” in *Proc. of 49th Annu. Meet. Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, pp. 1220–1229, 2011. [Article \(CrossRefLink\)](#)
- [18] 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. [Article \(CrossRefLink\)](#)
- [19] U. Hassan and E. Curry, “A capability requirements approach for predicting worker performance in crowdsourcing,” in *Proc. of 9th Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*, Austin, TX, USA, Oct. 2013, pp. 429–437. [Article \(CrossRefLink\)](#)
- [20] I. Boutsis and V. Kalogeraki, “Crowdsourcing under real-time constraints,” in *Proc. of IEEE 27th Int. Symp. Parallel Distrib. Process*, Boston, MA, USA, pp. 753-764, May 2013. [Article \(CrossRefLink\)](#)
- [21] E. Simpson and S. Roberts, “Bayesian methods for intelligent task assignment in crowdsourcing systems,” *Decision Making: Uncertainty, Imperfection, Deliberation and Scalability*, Basel, Switzerland: Springer, pp. 1-32, 2015. [Article \(CrossRefLink\)](#)
- [22] 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. [Article \(CrossRefLink\)](#)
- [23] V. Ambati, S. Vogel, and J. G. Carbonell, “Towards task recommendation in micro-task markets,” in *Proc. of Hum. Comput.*, pp. 80-83, 2011. [Article \(CrossRefLink\)](#)
- [24] M.-C. Yuen, I. King, and K.-S. Leung, “Task matching in crowdsourcing,” in *Proc. of 4th Int. Conf. Cyber, Phys. Social Comput. Int. Conf. Internet Things (iThings/CPSCOM)*, pp. 409-412, Oct. 2011. [Article \(CrossRefLink\)](#)
- [25] V. S. Sheng, F. Provost, and P. G. Ipeirotis, “Get another label? Improving data quality and data mining using multiple, noisy labelers,” in *Proc. of 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pp. 614-622, 2008. [Article \(CrossRefLink\)](#)
- [26] A. Brew, D. Greene, and P. Cunningham, “Using crowdsourcing and active learning to track sentiment in online media,” in *Proc. of 19th Eur. Conf. Artif. Intell. (ECAI)*, The Netherlands, Amsterdam, vol. 215, pp. 145-150, Aug. 2010. [Article \(CrossRefLink\)](#)

- [27] L. F. Capretz and F. Ahmed, "Making sense of software development and personality types," *IT Prof.*, vol. 12, no. 1, pp. 6-13, 2010. [Article \(CrossRefLink\)](#)
- [28] G. Kazai, J. Kamps, and N. Milic-Frayling, "Worker types and personality traits in crowdsourcing relevance labels," in *Proc. of CIKM*, Scotland, U.K, pp. 1941-1944., Oct. 2011. [Article \(CrossRefLink\)](#)
- [29] 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. [Article \(CrossRefLink\)](#)
- [30] Muhammad Zahid Tunio, Haiyong Luo, Cong Wang, Fang Zhao, Abdul Rehman Gilal, Wenhua Shao, "Task Assignment Model for Crowdsourcing Software Development: TAM," *Journal of Information Processing Systems*, 14(3), 621-630, March 2018. [Article \(CrossRefLink\)](#)
- [31] Muhammad Zahid Tunio, Haiyong Luo, Cong Wang, Fang Zhao, Abdul Rehman Gilal, Wenhua Shao, "Impact of Personality on Task Selection in Crowdsourcing Software Development: A Sorting Approach" *IEEE Access*, Vol. 5, pp. 18287-18294, 2017. [Article \(CrossRefLink\)](#)
- [32] 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. [Article \(CrossRefLink\)](#)
- [33] R. Valencia-Garcia, F. García-Sánchez, 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. [Article \(CrossRefLink\)](#)
- [34] 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. [Article \(CrossRefLink\)](#)
- [35] N.M.F. Qureshi et al., "A knowledge-based path optimization technique for cognitive nodes in smart grid," in *Proc. of IEEE Globecom 2018*, Abu Dhabi, UAE, pp. 1-6, Dec. 2018. [Article \(CrossRefLink\)](#)
- [36] I.F. Siddiqui, et al., "Edge-node-aware adaptive data processing framework for smart grid," *Wireless Personal Communications*, vol. 106. pp. 179-189, Mar. 2019. [Article \(CrossRefLink\)](#)
- [37] I.F. Siddiqui et al, "Pseudo-cache-based IoT small files management framework in HDFS cluster," *Wireless Personal Communications*, vol. 113. pp. 1495-1522, May. 2020. [Article \(CrossRefLink\)](#)



**Engr. Zainab Umair Kamangar** received her Bachelors of Engineering in Software Engineering from Mehran University of engineering and technology, Jamshoro, Pakistan, in 2015. Currently, she is pursuing her Masters of Engineering in Software Engineering from Mehran University of Engineering and Technology, Jamshoro, Pakistan. Her research interests are crowd-based software engineering, software testing, and machine learning.



**Dr. Isma Farah Siddiqui** is an Associate Professor in the Department of Software Engineering, Mehran University of Engineering and Technology, Pakistan. She received her Ph. D. degree in Computer Engineering with distinction from Hanyang University, ERICA, South Korea with the financial support of Higher Education Commission, Pakistan. She received "Best Ph.D. Graduate" Award from the college of Computing, ERICA, Hanyang University. She is reviewer of various renowned SCIE journals such as *Wireless Personal Communications*, *IEEE Access*, *Future Generation Computer Systems*, *KSII Transactions on Internet and Information Systems* and *Journal of Supercomputing*. She conducted various technical workshops and technical talks including IMTIC 2018 and 16th Annual symposium at MMC, Pakistan. She's been TPC of various national and international conferences including IMTIC'18 and FIIT'18. Her research interests include Smart Environment, Semantic Web, IoT and Big Data.



**Dr. Qasim Ali Arain** is an Associate Professor at Software Engineering, Mehran UET Jamshoro, Pakistan. He received Ph. D. in Information Security from Beijing University of posts and Telecommunication, China, through CSC scholarship. He was awarded the International Student best researcher award in all over china from Ministry of Education China and Best research student award from Beijing University of posts and telecommunication, Beijing on account of his research contributions and performance.



**Engr. Umair Ayaz Kamangar** received his Bachelor of Engineering degree with a major in Software Engineering from Mehran University of Engineering and Technology, Jamshoro, Pakistan, in 2015. Currently, he is doing Masters of Engineering in Software Engineering from the Institute of Information and Communication Technology, Mehran University of Engineering and Technology, Pakistan. His research interests are crowd-based software engineering, semantic web, and machine learning.



**Dr. Nawab Muhammad Faseeh Qureshi** is an Assistant Professor at Sungkyunkwan University, Seoul, South Korea. He received Ph. D. in Computer Engineering from Sungkyunkwan University, South Korea, through SAMSUNG scholarship. He was awarded the 1st Superior Research Award from the College of Information and Communication Engineering based on his research contributions and performance during studies. Currently, he is serving six guest editorials as Lead Guest Editor of Multimedia Systems, Springer, "Role of Deep Learning Models & Analytics in an Industrial Multimedia Environment", Guest Editor of Computers, Materials & Continua Special Issue "Artificial Intelligence and Big Data in Entrepreneurship", Guest Editor of Future Internet Journal Special Issue "Special Issue on Cyber-Physical Systems: Prospects, Challenges, and Role in Software Defined Networking and Blockchain" and Guest Editor of Internet Technology Letters "Special Issue on Deep Learning for Future Smart Cities", Guest Editor of Information Technology and Management Special Issue "Call for Papers on Disruptive Technologies and Innovations for Social Media Marketing" and Guest Editor of the special issue "Sustainable Energy Management in Intelligent Transportation," Also, he has served as General Chair Workshop NexGenRAN (Open-RAN: Open Road to Next Generation Mobile Networks) in IEEE Wireless Communications and Networking Conference (WCNC2020) 25th May, Seoul, South Korea and serving as Proc. Chair in Global Conference on Wireless & Optical Technologies 2020 (GCWOT'20) and General Chair Workshop Open-RAN: Open Road to Next Generation Mobile Networks in IEEE Globecom 2020, Taiwan. He is a reviewer of various prestigious journals such as Future Generation Computer Systems, Transactions on Emerging Telecommunications Technologies (ETT), Wireless Personal Communications (Springer), KSII Transactions on Internet and Information Systems, Journal of Supercomputing (Springer), IEEE Communications Magazine, IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics, IEEE Transactions on Industry Applications, IEEE Access, Mathematical Problems in Engineering (Hindawi Publishers), MDPI series of Journals including Symmetry, Electronics, Applied Sciences, Information, and Energies, Journal of Real-Time Image Processing (Springer). He has been a reviewer of various top-tier conferences such as IEEE Globecom2018, IEEE PIMRC 2017, IEEE ICACT 2019, AIIPCC2020, and IEEE ICACT 2020. He has been a TCP in IWWCN2017, CSA2017, IMTIC18, and WCSN2017 conferences and performed as session chairs with ICGCET Denmark, RTCSE19 USA, ICACT 2019 South Korea, and RTCSE 2020. He has evaluated several theses as external Ph.D. thesis evaluators. He has facilitated several institutes with Webinars on Big data analysis and Modern Technology convergence and served sessions with keynote talks on convergence with modern technologies. He is an active Senior Member of IEEE, ACM, KSII (Korean Society for Internet Information), and IEICE (Institute of Electronics, Information and Communication Engineers). His research interests include big data analytics, context-aware data processing of the Internet of Things, and cloud computing.