Improvisation Issues Of Developer Performance In Software Development Organization

Muhammad Usman Ashraf, Humaira Abbas, Iqra Ilyas, Mohammed Nawaf altouri, Hassan Mohammed Almalki, and Mousa Alalhareth

Abstract: Software development organizations (SDOs) are playing a vital role in bringing innovation, computerization, and smartness in emerging technologies. Since last two decades, software development organizations are increasingly faced by serval unforeseen effect challenges which requires off-the-cuff considerations and actions. Leading to developer performance, these challenges are including human effectiveness impact on software productivity, team flexibility, social processes of how software developer's work together, familiarity and different factors that influence productivity. Therefore, developer performance improvisation has been identified as one of underlying obstacle to overall team effectiveness. In this research, we have conducted a critical survey on improvisation issues of developer's performance in software development organizations, existing state-of-the-art methods to address these challenges. We finally discuss the future research directions that can be consider to address the improvisation obstacles at developer level in SDOs.

Index Terms: Software Development Organization (SDO), Developer performance, Improvisation.

1 Introduction

Software development is a process of creating new application that supports the needs of business and is organized in projects with fixed areas, budgets, and deadlines, commonly under the responsibilities of the software development functions. Many ways for developing software product can exist. IT operation and different hand is concerning running and dominant the varied code in production, monitoring and handling the IT infrastructure incidents and user requests. An IT operation exists in commonly within the software product center and includes such functions as service table, operation system and storage management, info and dealings management [1]. Why would like software development? Because software quickly turning a significant part of our society and there's no part of humane routine that has not been effected by software production. Business or industrial, diversion, civil, politics and education that everyone affected by software application and system. That's why software development becomes an important activity that requires be carefully studying, supporting, understand and improved throughout software development. The most intensive goal of software engineering community has been investigating and improved the problems of software development [2]. Software development is an important activity with the rise in software project. Software engineers often have less tool and techniques that is necessary for success of software project. Several aspects of development, is live with high degree of potency, automation and coarseness from code supply to testing bug the info will [3]. How to develop software and different phases of software development? The software development life cycle is a frame work for planning, structuring and executing tasks include with developing an information system. The software development includes different phases: initiation is the first phase identifies the problem that can be solved by software engineering and also managing the issue of scope and determines the functional requirements. The second phase is system thoughts development that is used to determine the scope of software application. The next step is planning this include acquisition arrange, thought of operations and project management arrange. After this need requirements analysis and then design the software application. And in development phase convert the designs from design phase into functional system. After the development phase perform the integration and testing the documents. A fundamental flow of software development has been presented in figure 1 as follows.

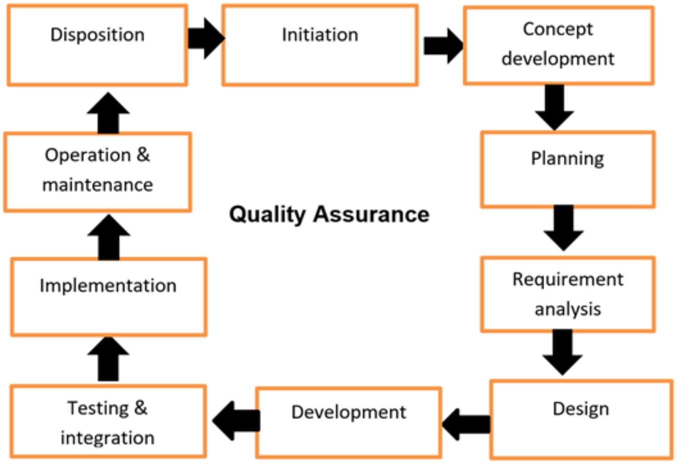

***Fig. 1**. Software Development underlying phases*

Implementation phase system goes to execution process and makes sure all requirements. Then perform operation and maintain the data and at the last disposition the software system [4]. Software development method has modified over time however continues to depend upon heavily on the utilization of groups. Building successful teams is a critical task that has received considerable analysis thought. Since success or failure depends on a team's ability to work to provide quality work and given the high charges of software project failure [5, 61]. Constant issues lead to developer performance problems increased and provide result decreased motivation for work, and inability to perform the development tasks. The most important question is how it is possible for developers to work under the performance issues and cannot remove these problems. However, needs to be somehow determining the high quality work. In other term, the research problem is to find and study the best condition for software developer work under the psychological pressure. For instance, when software developer works remotely operated in relax environment is moving to a desired location due to work in happy environment in the software organization. Rapidly increase pressure on software development, and mostly successful software project has attracted by many researchers and practitioners. Mostly software project consider fails and some project cancelled due to some issues mostly issues related to developer performance. Software development is an intricate problem involved different official and human organization factors. Team flexibility, unhappiness of developer, human effectiveness and productivity are the big issues of developer during development of software. That's why need to define and manage these issues for protection of software project. The primary objective of this research is ascertained the developer performance problems occur in software development organizations. Also find out the methods used to fix these issues through a critical review and evaluate the proposed approach to finds these issues. Developer quality is important for the software system there is needed to identify the developer issues. Rest of the paper is organized in such way that, section 2 presents the underlying developer performance attributes in software development. In section III, we have presented a list of challenges faced in achieving performance while software development. Section IV, describe the existing state-of-the-art methodologies use to addressed the determined issues and a comparative analysis. In section V, we have presented a comprehensive study discussion as well as our recommendations. Finally, we conclude the study in section VI.

2 DEVELOPER PERFORMANCE ATTRIBUTES IN SOFTWARE DEVELOPMENT

We have evaluated the developer performance attributes in software development. In this modern era customized especially software web application are given more importance because of global and simultaneously access. Software quality is more relayed with software developers and managers. There describe several numbers of quality attributes are efficiency, reliability, reusability, usability, correctness, testability, interoperability, portability and security [6, 58, 64, 65]. However, the quality of software product has not been confirmed that's why needs to evaluate the developer performance attributes. Performance attributes is the best assessment technique to validate the developer productivity and determine the way to enhance the software quality. Software system testability is associate degree external software quality characteristic that evaluates or estimate the hassle needed for testing and measures for software parts by process effectiveness and reusability notions [55]. Utilize of various parts is completed in part based mostly software's because of that once integration is done there are bound integration issues which are troublesome to handle [50]. Requirements for change request are important for product reliability. Development groups observant to write down performance moveable codes realize themselves with various choices, a trade-off between performance, movability and productivity [51, 57]. The foremost quality attribute is maintainability within the development of software system project. Maintenance is to contemplate the most effective activity for software system quality enhancements and optimization of software [49].

TABLE 1

Developer performance attributes in software development

Sr. no	Attributes	Description
1	Maintainability[49]	Maintainability is how different operation can be carried out effectively.
2	Reusability[50]	Create more complex system by using existing components.
3	Reliability [51]	Is an assessment tool produces stable and consistence results.
4	Usability [6]	Usability deals with user experience of software interface.
5	Flexibility [17]	Flexibility is an adaption in response to changes in environment.
6	Correctness [6]	Correctness is the ability of a software system to performing different tasks in an accurate manner.
7	Interoperability[53]	Is the ability two or more components or system exchange information and used exchanged information.
8	Portability [54]	How efficiently a scientific application available resources.
9	Testability [55]	Software testability is an ease of facilitating testing process.
10	Security [6]	Prevention of unauthorized access and malicious attack.
11	Efficiency [56]	is a set of attributes define relationship between the level of performance of the software and the amount of resource used.

3 CHALLENGES TOWARDS DEVELOPER PERFORMANCE

During the development process, software developers' face several challenges that have different effects on software project. More important to visualize the developer's challenges for improving the developer productivity and reduce the factors of performance. Challenges of software development mostly cause the failure of software product need to carefully understand the issues and determine the factors of these issues. Software developers' challenges are innovative behavior aims at generating, promoting and useful ideas about software products, processes or procedures at work [40]. Software development depend on specific team and team manger divide the task process between each team member and execute each phase of project. Project divided into different phase but mostly issues exist in new team members who do not have more experience than older team members [6]. They gathered information from the respondent in one file organized by cluster and respondent. A general assumption is that temperament characteristics considerably have an effect on the human behavior. Though most of the people agree that sorts of behaviors vary in numerous things, for many individual there's associate noticeable pattern of consistency in their behaviors [23]. Totally different aspects of the productivity programmers are evaluated. Elaborated factors that have an effect on software system productivity are known on the idea of surveys, interviews and observation of computer programmer and project managers. A number of them embody computer programmer expertise.

TABLE 2

Attribute based developer performance challenges

Challenges	Description
Self-rated productivity [16]	Developers who have completed their task selves and spend freed up time on other helpful activity.
Team or personality conflict [23]	Software project fails due to social issues such as team or personality conflicts.
Lack of communication[34]	Communication is a main challenge for team effectiveness.
Humane effectiveness [18]	Human Effectiveness, i.e., the emotional state of a person.
Stressful environment [42]	In stressful environment affect the mental state of developers.
Age stereotypes [21]	Hinder the trust, communication, and knowledge exchange.
Emotional intelligence	Emotional intelligence effect on the trust, team

[43]	performance and create conflict.
Long working hours [22]	Have been the suspected risk factor for depressive symptoms.
Social interaction[1]	A situation where the behavior of one actor influence by other actor.
Impact of organization culture [48]	Organization culture has impact on employee proactivity.
Unhappiness[38],[39]	Unhappiness causes the psychological disorders.
Psychological safety [44]	Psychological safety should be treated as a group-level concept estimating team climate.
Team norm[44]	The coworkers should be behaving in a specific way.
Speed of task resolution [37]	Speed of task execution can be changed during development of project.
Task type[37]	Task type must be categorized when divided among the team members.
Neglect to update legacy [36]	Software developer ignores the updating of software dependencies.
Barriers of knowledge sharing [19]	Knowledge sharing barriers cause the inefficiency in team communication.
daily life of software developers[45]	Software developers spend little time on development.
Human factors [33]	Humane factors include emotions, effective states, and moods.
Optimal group size [20]	Development team face problem if team size is too small or too large.
Effect of work environment [15]	The physical work environments have effect on their satisfaction and ability to get the work done.
External dependencies[47]	External coordination entails how teams coordinate with their environment.
Difference between task process [6]	It could account the difference between two groups of productivity.
Software engineers nature[12]	Software engineers nature changed need to re-look the way that traditionally
Innovative behavior[40]	Software developers' innovative behavior is imperative for the success
Role perception [40]	Role perceptions includes i.e., role ambiguity and role conflict

4 EXISTING STATE-OF-THE-ART METHODS

With the passage of time the massive demand software development has increased in a highly and dynamic market and mostly successful software project attracted by researchers and developers. They determined the factors that increased the success of software development by investigating the influence of team flexibility on software project output. However, in current section we present different state-of-the-art methods to overcome the issues in software development as discussed below. Günsel, A, et.al (2012) [17] explained a framework for flexibility of software developer team in project management and software development project. They emphasized on the team flexibility has become a critical issue in software development organization. They examine the relationship between team flexibility and outputs of software projects: market success, speed to market and the functionality of new software product. The study explores 2 dimensions of team flexibility are team autonomy and team diversity has positive or negative effects on the outputs of software projects. The sample size was comparatively small (n=86), and data used in this research obtained from single informant from a given project. The study was conducted in specific national context, TURKEY; the researchers should be alert determining the results of different cultural contexts. Although, they enhanced the market success of the software products, develop them faster, and increased their functionality, management should enhance the flexibility of the project team. They used PLS path modeling to measure the latent variables of team flexibility are team sovereignty and team variety with products. The result shows that team autonomy is positively related to the market success (β= .36, p<.01), the speed (β=.31, p<.01) and the functionality (β=.28, p<.01) of new product. They found the result of second dimension team diversity is positively influenced on the speed (β=.20, p<1) and the functionality (β= .35, p<.01) of new product. However, Hypothesis 1 fully supported and 2 were partly supported. The team flexibility sub dimension team autonomy and team diversity explain 21% of the variance, in the market success of the software development project, 29% of the variance in the market speed and 37% of the variance in the functionality of the new software product. The study determined the factors that increase chances of success of software development project by exploring the effects of team flexibility. In future research they find the area of teamwork, cross-functionality and co-development. A.N. Mayer et.al (2014) [32] proposed a framework for perception of productivity of software developers in software development organization. They focused on the software developers' retrospection and improved the productivity through the development of new tools and more practices. The suggested approach investigates the software developers' perception of software development productivity. They also described the interruption and context switching cause the failure of software development productivity. They found developers perceive more days as productive when they complete their more or big task without interruption and context switching while developing software project. They used two methods to find the results online survey and observation. In online survey may not be represent the general population of software developers and the generalizability of this survey limited. In observation method small number of population participates. Although, in this research find productive days of software developers and also finds productive and unproductive activities as of software developers while developing software project. Fuggetta, A., and Di Nitto, E. (2014) [2] introduced a framework for software process and software development becomes critical activity that needs to be carefully understood, studied and supported. They focused on complex challenges for software engineering researchers and practitioners. Over the years, complexity and size of software have enormously increased and has become a critical asset of society. This increasing important has placed a lot of challenges and pressure on software development teams. In the past decade, the approach of software development has become holistic and multidisciplinary addressing the broader range of issues. This transformation demands openness to encounter the challenges. The trends affecting the society need to be "open". Researchers should abandon their comfort zones and to be open to new challenges, suggestion, ideas and trends emerging in the society. As the world has changed they were supposed to imagine and anticipate those changes accordingly in order to shape our future. A key element of computer science and software engineering can help them in this work of software process community. Graziotin .D, Wang .X and Abrahmsson .P (2014) [33] explained a comprehensive study for problems of software developers in software development organization. They improved developer's productivity and software quality by focused on people and provide some effects that make software developers satisfied and happy. They described human factors, affective states, emotions and moods that affect the development process and performance of developers. There required an analytical problems solving skills and creativity is more important while developing software and define a relationship between affective states, creativity and analytical problems solving skills of software developers. Also have some limitation in this research, this study limited in software development experienced compared with professionals and also employment of a median spilt to compose the groups. Although in this research, determined that happiest software developers are significant for analytical problems solvers and shows that creativity needed more research on it. Ortu, M et.al (2015) [18] described an architecture for issue fixing time and human effectiveness in software development organization. They focused on the software quality and improve developers' productivity in software development organization. The suggested model analyzed the mood of developer teams, allowing the project leaders to anticipate and resolve the potential threats to productivity in the community. They highlight the impact of effectiveness on productivity in the form of issue fixing time. They discussed the three important metrics i.e. emotions, sentiments and politeness were independent and have weak co-relation with the control metrics. Effectiveness metrics statistically improved the exploration of model of issue fixing time, as compared to model based on control metrics. In other words, comments containing love or joy have shorter fixing time, while comments containing sadness emotion have longer fixing time. And those containing politeness has shorter fixing time no matter they have negative politeness or positive politeness. The purposed approach logistic regression model used different measurements units to calculate these three affective metrics. First calculate the correlation between three metrics and measure the politeness, sentiments and emotions of developers' comments. Measure the polite value use different rules and normalize the average issue politeness in arrange from 0 to 1. Politeness issue distinguished with three groups of issue low, mediums and high politeness issues. Low and high politeness have shorter fixing time containing respectively 38.8% and 10.4% the total number of issues. Then calculate the sentiments, the sentiments used sentistrenght. Its value in arrange from -5 to 5 and normalize again in a range from 0 to 10 means extremely impolite and negative, 0.5 means neural politeness and sentiments and 1 means polite and positive. Then calculate emotions, average sentiments and politeness of about 560k comments (about 68k issues). Nakata A. (2017) [22] described a community based and cross sectional study for investing long working hours that increased the risk of depressive symptoms and reduced job satisfaction in small and medium scale business. They focused on the depressive symptoms that cause the working impairments and reduced productivity of developers and improved job satisfaction among those long working hours. They found independent relation between working hours and job satisfaction with depressive symptoms but when tested the job satisfaction and working hours then find job satisfaction is the main cause of depressive symptoms. 296 small and medium businesses were surveyed using self-administered questionnaires to evaluate the job satisfaction, working hours, depressive symptoms and covariates. The Center for Epidemiologic Studies Depression Scale (CES-DA) was used to assess depressive symptoms. The limitation of this study, collected data is cross sectional and use self-report to assess the working hours, job satisfaction and depressive symptoms they not use objectives measure or diagnostic criteria. Although in this research, combined different working hours of employees with high or low job satisfaction and create high or low risk of depressive symptoms. Authors described collective association of working hours and job satisfaction on depressive symptoms assess through DES-D>16 OR 95% CI. Work 6 to 8h with high job satisfaction then DS=1.00 and when with low job fulfillment DS=1.48. If work >8 to 10h with high job satisfaction DS= 0.93 and with low job satisfaction DS=2.21. Work > 10h with high job fulfillment DS= 1.39 and with low job satisfaction DS=2.39. The current study produced three main finding: first, long working hours, but those working hours 12+hrs were weekly but significantly related with increased risk of depressive symptoms. In future work, they should be determined the working hours, job satisfaction and depressive symptoms assess through objectives measures and diagnostic criteria and give long time to respondent for answering the questionnaires and also include other factors in future work. Licorish S, A and Macdonell S, G (2017) [37] presented a mechanism for exploring software developers work practices in software development organization. Authors emphasized on practices, and improved the speed of task resolution, find task difference and participation and engagement increased in software development organization. They found software development become a challenging activity for successful software need to create members of team work collectively and find different factors that influence speed of task resolution. The suggested approach finds that behaviors and intrinsic issues related with extrinsic factors that are becoming more significant predictors for speed of task resolution. This research provides number of insight but also have some limitation, the communication can have measured by messages that cannot represent all project team. Direct content analysis has interruption and cultural difference directly affect the software development performance and collect studied date from single organization. They followed questionnaires approach to find the results about work practices and collect data from one organization. RQ1 of more than 30,000 software task get from Jazz repository there defect the largest group, comprising 23,331 tasks, 12% were classified as support tasks and the lasting 11.6% were categorized as enhancement. The result of RQ2 of the 117,101 e-mails that exchanged all over the 30,646 task, defects on working by exchanged the 88,874 messages by experts, employed on enhancement features exchanged 12.4% and 11.7% messages exchanged working on support task. In RQ3 described the different variance of speed of task execution. When the software designers working on enhancement features than practitioners intensively engaged and exchanged 4.1 messages for each task. Although, in this research finds that practitioners studied involved with fixing defects and enhanced the task resolution speed and need more and more practices to done this work. In future work communication cannot asses through messages and use date from two or more organizations. In future study, explore new features and software enhancement and observe more factors that influence speed of task resolution. Graziontin et.al (2017) [38] explained the consequences of unhappy developers in software development organization. Authors focused on the issue of developer performance of unhappiness and increase happiness among software developer. They identified 254 codes related to the consequences of unhappiness, which result in 49 types of consequences but there described some most frequent types due to space limitation. In this research, they found the highest effect of unhappiness on productivity and performance, by gathering the categories of low performance and process related productivity. They found the issues of unhappiness related to mental health such as low self-esteem, high anxiety, burnout and stress. Critical issues are depression and low motivation among developers. But these issues can be investigating through psychology research. These classification schemes link the causes and effects of happiness and it will be used as a suggestion for developers and managers at work. The study was distended the previous gathered data by suggesting that limiting unhappiness can limit the injury in terms of many factors from individual, artifacts and method perspective. In future work they determined the remaining codes and reduce the mental health problems with developers that create negative effects on software development. Wang, Y. Y., Davison, R. M., and Huang, Q. (2017) [40] elaborated a framework to understand how software developer innovates in an IT enabled work environment by considering their role perception. They focused on the innovative behaviors of software developers and understand who their role responsibilities are more likely innovate, while who disagree their role responsibilities innovate less. The suggested approach, explained of how IT affordances were related to software developers' role perception and innovative behaviors and open new doors of innovation. Authors measured, and analyzed the collected data and measurements items are adapted from current literature. First, they adapted the measures for individual innovation from the work of Tierney and Wang et al. Through these measurements identified the opportunities for new product, quality of service [62] and process. Secondly measure the role perception by item adapted from Rizzo et al. Sample item for RA was "I know what my responsibilities are". Sample item for RC was "I receive incompatible request from two or more people". IT affordance will be measure based on the work of Chatterjee et al. At last measure the control variables add firm size, team size and demographic information like software developers' age, gender, educational background and work experience. In further work, they can be finds the latest technologies to solve these issue. Rezvani, A., and Khosravi, P. (2019) [46] proposed a framework for mitigating the stress and fostering trust among software developer in information system project. The primary focus of this research was on the success of information system project and decrease stress and increase trust among software developer and determined the impact of developers' emotional intelligence on stress, trust and performance. Developers give short time to enhance their personal skills and ability that can overcome these issues. According to this study, emotional intelligence of software developers control stress and create trust among software developers and increase the developers' performance. Results of these problems presented through 8 hypotheses. In first hypothesis suggest that emotional intelligence of software developers is positively related with the performance (β=.31, p<.001). Hypothesis 2 present the negative association between software developers emotional intelligence and stress (β=0.40, p<.001). in third hypothesis shows that emotional intelligence was positively associated with trust (β=.42, p<.001). Hypothesis fourth predicted trust was positively related with performance (β=.14, p<.05). In fifth hypothesis the negative association with software developers stress and performance (β=-0.18, p<.01). Hypothesis describe the effect of role ambiguity (β=.15, p<0.001). Role Ambiguity defines the relationship among stress and performance. Through this study define how emotional intelligence affects the capability to manage, understand and switch the positive and negative emotions. In future research they can be defined problems with other participants of stress and trust between software developers such as, project leaders or team members and present other representative in place of role ambiguity use the culture to increase or reduce the effect of stress. Jalote, P., and Kamma, D. (2019) [6] proposed a general framework for studying the task processes in software development organization. They emphasized on the task processes and improved the programmers' productivity in software development organization. The suggested framework applied on some live projects and identified the 2 groups of programmers: high productivity and average productivity programmers and analyzed the difference between 2 groups' high and average productivity. In this work is that the experiments were completed in one company and in one country, but they did not show interest in using the framework being used by one company in theirs. Before transfer the high productivity programmers to average productivity programmers, first they identified some important steps that need to be change, add or delete and then trained the programmers on those. Shows that increase the similarity between task process of high and average productivity programmers. Moe, N. B et.al (2019) [47] described a framework for team autonomy in large scale agile. They focused on the barriers of team autonomy and solve the problems in large scale development. During development, team needs to line up with other teams and the rest of the organization. However, agile team faced different challenges in large scale development. Through case study of three projects identify the two barriers of team autonomy that are overall direction and external dependencies. They found the goals that can set by management team members can't involve in it that's why mostly team members don't know what the goals are? On large scale development need to understand the shared direction and established team struggle with shared direction which result in specific team members protect the team from external noise. In this research focused on the barriers that can reduce the responsibility and authority of the team autonomy. They determined how the team coordinates and communicate with other team and other company. Reduce the responsiveness and flexibility for their many part of work such as planning [63], scheduling and assigning the task to the workers. In further work, they can be defined the shared direction and focused on investigating the barriers with autonomous teams in large scale agile for example related to reducing dependencies between teams. Johnson .B, Zimmermann .T and Bird .C (2019) [15] described a framework for the effect of work environment in development organization. They focused on the productivity of developers and understand the factors of work environment that affect the productivity and job satisfaction. They found several factors that as important for work environment are personalization, social norms and signal, room composition and atmosphere, environment affordance, work area and furniture. They used different survey for satisfaction and productivity of software engineers and compared the collected for developers on the Program Management, IT Operations, Marketing and Business

Program and Operations discipline.

TABLE 3
Collected data in the research

Population	Invited	Responded
Software Engineering	1,591	298(18.7%)
Program Management	548	131(23.9%)
IT Operations	546	162(29.7%)
Marketing	516	103(20.0%)
Business Program & Operations	467	142(30.4%)

4.1 Challenges based comparative Study

Now, we analyze the developer performance issues or challenges according to the proposed solutions and evaluate its significance based on different attributes. We indicate what the methods suitable for selection of challenges are and find the psychological measurements of developer state of mind because the developer factors occur in software development organization [42]. In applied math analysis studied comprehensive, practitioners' massages that were contributed in relation, arbitrarily elect software package tasks, employing a directed content analysis approach [31, 60]. In empirical approach conduct review on the existing literature review and also used survey instrument to determine the issues [29]. Analysis to investigate the information from the semi structured interviews performed with team members, a way for characteristic analyzing and coverage themes found in qualitative knowledge that are relevant to explain a development. In interview technique are often conduct depth interview on some target firm and modify questioners [26]. With respect the identified challenges, a comprehensive comparative analysis of existing methods has been conducted and presented as shown in table 4.

TABLE 4
Challenges based comparative analysis of existing proposed solutions.

Challenges	Existing Proposed solutions									
	[19]	[48]	[18]	[42]	[43]	[15]	[37]	[44]	[16]	[6]
lack of communication	✓	×	×	×	×	×	×	×	×	×
Innovative behavior	✓	×	×	×	×	×	×	×	✓	×
Self-rated productivity	×	×	×	×	×	×	×	×	✓	×
Humane effectiveness	×	×	✓	×	×	×	×	×	×	×
Stressful environment	×	×	×	✓	✓	×	×	×	✓	×
Age stereotypes	×	×	×	×	×	×	×	×	✓	×
Effect work environment	✓	×	×	×	×	✓	×	×	×	×
Emotional intelligence	×	×	×	×	✓	×	×	×	✓	×
Long working hours	×	×	×	×	×	×	×	×	✓	×
Impact of OGC	×	✓	×	×	×	×	×	×	×	×
External dependencies	✓	×	×	×	×	×	×	×	×	×
Unhappiness	×	×	×	×	×	×	×	×	✓	×
Psychological safety	×	×	×	×	×	×	×	✓	✓	×
Team norm	×	×	×	×	×	×	×	✓	✓	×
Neglect to update legacy	×	×	×	×	×	×	×	×	✓	×
Barriers of KS	✓	×	×	×	×	×	×	×	×	×
Task type	×	×	×	×	×	×	✓	×	×	×
Speed of task resolution	×	×	×	×	×	×	✓	×	×	×
Daily life of developer	✓	×	×	×	×	×	×	×	✓	×
Humane factors	×	×	×	×	×	×	×	×	✓	×
Difference between TP	×	×	×	×	×	×	×	×	×	✓
Team or personality traits	×	✓	×	×	×	×	×	×	×	×

4.2 Analysis of challenges based on software development attributes

In this section analyze the challenges that effect the performance quality attributes. If non-functional requirements consider fails within the early stages of software package development might end in poor quality software, augmented maintenance prices and time [7]. That's why need to examine the challenges of developer's performance in software development organization. Absence of documentation for security, restricted quantity of your time for testing security in sprints, and issue for group action security connected activities are major security problems with developer performance [6]. Consequently, there has been analysis for integrating, planning and managing the developers' performance. The unhappiness of developers has negative impacts on many vital software package engineering organizations. Productivity and performance are the aspects that suffer most from sad developers [38].

TABLE 5
Analysis of determined challenges based on software development attributes.

Challenges	Existing Proposed solutions										
	[50]	[51]	[6]	[6]	[6]	[53]	[54]	[55]	[56]	[17]	[49]
personality traits	×	✓	×	×	×	×	×	×	×	×	✓
Self-rated productivity	×	×	×	×	✓	×	×	×	✓	×	×
Age stereotypes	×	✓	×	×	×	×	×	×	✓	×	✓
Long WH	×	×	×	×	×	×	×	×	✓	×	×
Effect of WE	✓	×	×	×	×	×	×	×	×	×	×
Barriers of KS	×	✓	✓	×	×	×	×	×	×	×	×
Humane effective	×	×	✓	×	×	×	✓	×	×	×	×
Optimal group size	×	×	×	×	×	✓	×	×	×	✓	✓
Less communication	×	×	×	×	×	×	×	×	✓	✓	×
External dependencies	×	×	×	×	×	×	×	×	×	✓	×
Stressful environment	×	×	×	×	✓	×	×	✓	✓	×	×
Unhappiness	×	×	×	×	×	×	✓	×	✓	×	×
Psychology safety	×	×	×	×	×	×	×	×	✓	×	×
Team norm	×	×	×	×	×	×	×	×	✓	×	×
Innovative behavior	×	×	×	×	×	×	×	×	×	×	✓
Social interaction	×	✓	×	×	×	×	×	✓	✓	×	×
Neglect to update	×	×	×	×	✓	×	×	×	✓	×	×
Difference between task process	×	×	×	×	×	×	×	×	✓	×	✓

[49] Maintainability, [50] reusability, [51] reliability, [6] correctness, usability and security, [53] interoperability, [53] portability, [55] testability, [56] efficiency and [17] flexibility.

Table 5 presents analysis of challenges and attributes. The first challenge team personality conflicts that support the quality attributes reliability and maintainability. If the developer behavior is not good, then personality is not better and cannot manage the software project and product is not consistent [23]. The fourth row hold the next challenge is self-rated productivity, if software workers is not depending on other worker then create self-rated productivity. If developer depend on other workers, then software project loose the efficiency and security cannot be maintaining [16]. Innovative behavior is a type of extra role performance behavior that smoothest the operational of organizations but requires individual's unpaid effort outside formal job requirements and important for maintainability of software project. Basic unit of sociological analysis is not action but reaction [40]. Thus, the social interaction is the elementary process in all social organizations and support the reliability of product and efficiency is maintaining if interact with each other [1]. Some key variations were found between the task processes, that might account for the distinction in productivities of the 2 teams that drawback have impact of efficiency and maintainability [6]. Long operating hours are suspected to be risk factors for depressive symptoms; it's not well understood the circumstances under which long working hours were related with it [22]. Software package development groups may suffer new barriers higher than and the far side the well-known culture and distance based challenges. These problems related cultural and age diversity [21]. Affective analysis, measuring emotions, sentiments and politeness, apply to developer issues report, can be useful to identify and control the mood of the development team. Developing software package systems could be a data intensive task, and intrinsically is heavily influenced by the state of mind of developers [42].

5 Discussion and Recommendations

Software development organization face different issues of developer performance these issues cause the failure of software project. Consequently, software development becomes an essential activity that must be fastidiously perceive understand and supported. We have determined developer performance issue or challenges and analyze the best approach use to solve these issues. Due to these issues affect the developer productivity and software quality. Developer productivity is the important module of software project, if developers give input to develop software but in response don't provide output conferring to its inputs its means efficiency of developer is reduce. In past few years ago, most of the projects consider fails and some cancelled due to developers' performance issues. These issues are unhappiness among software developers, co-workers don't share their knowledge and familiarity with each other, and newcomers in development field take more time to complete a process, age difference between employees most younger employees not have more experience than older employees, sometimes the development team is too large or too small it also create disturbance among developers. If development environment is stressful then developers not focus on the work. To address these issues, find some techniques or framework used and which techniques are best for these issues. In current research, we have calculated the developers' performance issues in software development organization through a critical review. First of all, we defined some attribute or parameters to enhance the developer performance in software development. The attributes are maintainability, testability, reliability, usability, reusability, flexibility, portability, interoperability and correctness. Maintainability is modification of software product, testability is evaluating the product complexity, reliability is make sure all critical system is safe, in reusability create more complex system by using existing components and usability is relating with ease of use and ease of learning, flexibility is an adaption in response to changes in environment, performing different task in accurate manner is correctness and portability is efficiently [66, 67] used the scientific application. Interoperability is the two or more components [59] or system exchange information and used exchange information. We evaluated the performance portability metric for developer performance from 2 real application codes, using both application efficiency and architectural efficiency. Requirements gathering are the crucial activity in which we have tendency to gather the requirements of the software developer that need to be carefully studied. By following developer performance factors, the challenges should be addressed: increase happiness among developer because happy software developers solve problems better, developer productivity is a big problem in software development; their social interactions are hampered by many factors. Humane effectiveness is the emotional state of person if is not maintain that break the team ability to succeeds the software product. Some challenges are self-rated productivity, lack of communication; neglect the update dependencies, difference between task process and nature of software engineers' change and manage the team size in software development organization. Psychological safety is main problem, more important to have knowledge about developer psychology and clear the team norm. Developers' performances are the important components for the success of a software development project, this is not primary important for execution of task, but permanently and core of the whole development organization. Communication in distributed setting is the main challenges to team effectiveness. Software developer innovative behavior is imperative for the achievements of organizations innovation. Politeness level of communication may facilitate developers and managers to watching the mood of workers in development companies. This study considers the 2 forms of quality attributes reliability and efficiency contributes to software development team flexibility that in turns impacts the standard of the system developed. Leadings to these challenges some techniques or approaches used: mostly used the survey method to determine these issues. In survey techniques used the interviews, questioners and observation method to address these issues. They also used fellow up survey and paper based survey to identify the challenges. For different issues used the PSL path modeling, neuroimaging techniques, logistic regression model, multiple linear regression model, general framework, empirical approach and quantitative analysis approach used to report these challenges. Empirical approach improved the ability to make or configure more effective team in software development companies. They used the partial least squares approach to estimate the measurements and structural parameters in structural equation model. They conducted depth interviews with employees of some target firms will be undertaken to understand the research context and amend the questioners. Questioners are the best approach to address the challenges of developers' performance in software development. Ultimately, through a critical analysis of techniques to identify the best method that provide the solutions of these factors of software developer face during software development. For identification of challenges of lack of communication, innovative behavior, effect of work environment, knowledge sharing barriers and daily life of developers used the interviews method. The factors are stressful environment, long working hours, unhappiness, psychological safety, team norm, personality conflicts, ignore the need of update, age differences, innovative behaviors and self-rated productivity support the questioner's methods that's why is the best approach to determine these issues. Age stereotypes used the quantitative survey; this technique provided the possibilities to study a large number of participants, which increase self-confidence in the reliability and validity. For stressful environment they used the neuroimaging techniques such as fMRI to understand the mind conditions of developers. In the end described the analysis of attributes based challenges in table 4 what challenge has Impact on the quality attributes of performance. Interoperability can be disturbing if worker group size is not mange, when don't communicate with each other and external dependencies has impact on flexibility. When evaluating the maintainability firstly solve the developer performance issues. Quality attribute efficiency cannot be achieving if developer work in long hours, age difference, when developer unhappy then reduce efficiency and when ignore the update dependencies and difference between the task process cause the negative effect on product efficiency. After a critical review, we make following recommendation to the policy makers and mangers of organization. Creates a favorable work condition in the organization and create leadership roles among employees. Also arrange proper training for improving performance and reduced workload of employees by using stress management. To fulfill these recommendation conduct survey on software developers. We should be conducted depth interviews on software developer, its sample populations are developers work in software houses and software organizations. This method is helpful to identify the developer performance challenges and provide a relax environment to developer for development of different software project.

6 CONCLSIONS AND FUTURE WORK

Some programmers are much more productive compared with others. In this research, we studied the impact of developers' factors on software productivity and quality. In this review study strives to comprehensively determined, appraise and synthesize all the relevant studies on these problems. To collect all existing material that matches some pre-specified issues of developer performance in software development. During software development organizations face different kinds of challenges; these challenges associated with developer productivity and software quality. Developer performance issues cause the failure of software project that's why need to carefully studied these issue such as unhappiness among developer, age difference, stressful environment, team autonomy barriers, increase trust and reduced stress and emotional intelligence is additionally to enhance the team performance and productivity. Different attributes used to enhance the team performance and reduced the developer performance issues these are: maintainability, reliability, privacy [68], novel programming skills [69, 70] reusability, flexibility, usability and correctness. Through a critical review we analyzed different approaches to determine these issues survey method mostly used conduct depth interviews and amend questioners on software developer of different software organization and then used PLS path modeling to measure the result. Researchers used fMRI techniques to understand state of mind of software workers. Developer performance issues exist when software development process is start because successful software project depend on the abilities on developers that's why need to solve these issues. It uses specific technique that chooses with the aim of minimizing the inherent bias, and notices the source of enhancing reliability. In further, researchers should focus on effect of work environment; virtual workplace and communication channels and make possible people talk less and email more. Reduce dependencies between team members and need to better understand the impact of task process in other organization and find the difference between task process in which programmer give more attention to modeling, coding, bug fixing and enhancement. And among software developer flourishing happiness and satisfied by making the work days good and daily life of software developer spend more time on development and improved job satisfaction and developer productivity. Should be focus on how gender relate with developer's productivity.

ACKNOWLEDGMENT

The authors wish to thank all participants from different countries who voluntarily share the required information during the survey phase.

REFERENCES

[1] Iden, J., & Bygstad, B. (2018) the social interaction of developers and IT operations staff in software development projects. International Journal of Project Management, 36(3), 485-497.

[2] Fuggetta, A. & Di Nitto, E. (2014, May) Software process: In Proceedings of the on Future of Software Engineering (pp. 1-12) ACM

[3] Buse, R. P., & Zimmermann, T. (2012, June). Information needs for software development analytics. In Proceedings of the 34th international conference on software engineering (pp. 987-996). IEEE Press

[4] Lemke, G. (2018). The Software Development Life Cycle and Its Application.

[5] Anderson, G., Keith, M., Albrecht, C., Spruill, A., & Pettit, C. (2019, January). Optimizing Software Team Performance with Cultural Differences. In Proceedings of the 52nd Hawaii International Conference on System Sciences.

[6] Jalote, P., & Kamma, D. (2019) Studying Task Processes for Improving Programmer Productivity. IEEE Transactions on Software Engineering

[7] Behutiye, W., Karhapää, P., Costal, D., Oivo, M., & Franch, X. (2017, November). Non-functional requirements documentation in agile software development: challenges and solution proposal. In International Conference on Product-Focused Software Process Improvement (pp. 515-522). Springer, Cham

[8] Alzoubi, Y. I., Gill, A. Q., & Al-Ani, A. (2016). Empirical studies of geographically distributed agile development communication challenges: A systematic review. Information & Management, 53(1), 22-37.

[9] Li, Y., Chang, K. C., Chen, H. G., & Jiang, J. J. (2010). Software development team flexibility antecedents. Journal of Systems and Software, 83(10), 1726-1734

[10] Shahzad, F., Xiu, G., & Shahbaz, M. (2017). Organizational culture and innovation performance in Pakistan's software industry. Technology in Society, 51, 66-73.

[11] Jan, S. R., Dad, F., Amin, N., Hameed, A., & Shah, S. S. A. (2016). Issues in global software development (communication, coordination and trust)—a critical review. training, 6(7), 8

[12] Sharma, V. S., Mehra, R., Kaulgud, V., & Podder, S. (2018, May) An immersive future for software engineering: avenues and approaches. In Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results (pp. 105-108). ACM

[13] Ali, N., & Lai, R. (2016). A method of requirements change management for global software development. Information and Software Technology, 70, 49-67.

[14] Zhang, Y., Wang, H., Yin, G., Wang, T., & Yu, Y. (2017). Social media in GitHub: the role of@-mention in assisting software development. Science China Information Sciences, 60(3), 032102

[15] Johnson. B, Zimmermann. T & Bird. C (2019) the Effect of Work Environments on Productivity and Satisfaction of Software Engineers. IEEE Transactions on Software Engineering

[16] Murphy-Hill, E., Jaspan, C., Sadowski, C., Shepherd, D., Phillips, M., Winter, C., ... & Jorde, M. (2019). What Predicts Software Developers' Productivity?. IEEE Transactions on Software Engineering..

[17] Günsel, A., Açikgšz, A., Tükel, A., & Öğüt, E. (2012) The role of flexibility on software development performance: An empirical study on software development teams. Procedia-Social and Behavioral Sciences, 58, 853-860

[18] Ortu, M., Adams, B. Destefanis, G., Tourani, P., Marchesi, M., & Tonelli, R. (2015, May) Are bullies more productive?: empirical study of affectiveness vs. issue fixing time. In Proceedings of the 12th Working Conference on Mining Software Repositories (pp. 303-313) IEEE Press.

[19] Ghobadi, S., & Mathiassen, L. (2016). Perceived barriers to effective knowledge sharing in agile software teams. Information Systems Journal, 26(2), 95-125

[20] Bhowmik, T., Niu, N., Wang, W., Cheng, J. R. C., Li, L., & Cao, X. (2016). Optimal group size for software change tasks: A social information foraging perspective. IEEE transactions on cybernetics, 46(8), 1784-1795.

[21] Schloegel, U., Stegmann, S., Van Dick, R., & Maedche, A. (2018). Age stereotypes in distributed software development: The impact of culture on age-related performance expectations. Information and Software Technology, 97, 146-162

[22] Nakata, A. (2017). Long working hours, job satisfaction, and depressive symptoms: A community-based cross-sectional study among Japanese employees in small-andmedium-scale businesses. Oncotarget, 8(32), 53041

[23] Yilmaz, M.,O'Connor, R. V., Colomo-Palacios, R., & Clarke, P. (2017).An examination of personality traits and how they impact on software development teams. Information and Software Technology, 86, 101-122

[24] Tams, S., & Dulipovici, A. M. (2019, January). The creativity model of age and innovation with IT: How to counteract the effects of age stereotyping on user innovation. In Proceedings of the 52nd Hawaii International Conference on System Sciences.

[25] Magni, M., Maruping, L, & Hoegl, M. (2008) Improvisation and performance in software development teams: The Role of geographic dispersion. ICIS 2008 Proceedings, 154

[26] Espinosa, J. A., Slaughter, S. A, Kraut, R. E., & Herbsleb, J. D. (2007). Familiarity, complexity, and team performance in geographically distributed software development. Organization science, 18(4), 613-630

[27] Fagerholm, F., Ikonen, M., Kettunen, P., Münch, J., Roto, V., & Abrahamsson, P. (2015). Performance Alignment Work: How software developers experience the continuous adaptation of team performance in Lean and Agile environments. Information and Software Technology, 64, 132-147.

[28] Ghayyur, S. A. K., Ahmed, S., Ali, M., Razzaq, A., Ahmed, N., & Naseem, A. (2018). A Systematic Literature Review of Success Factors and Barriers of Agile Software Development. International Journal of Advanced Computer Science and Applications (IJACSA), 9(3).

[29] Zafar, A. A., Saif, S., Khan, M., Iqbal, J., Akhunzada, A., Wadood, A., ... & Alamri, A. (2018). Taxonomy of factors causing integration failure during global software development. IEEE Access, 6, 22228-22239.

[30] Mäder, P., & Egyed, A. (2015). Do developers benefit from requirements traceability when evolving and maintaining a software system?. Empirical Software Engineering, 20(2), 413-441.

[31] Licorish, S. A., & MacDonell, S. G. (2018). Exploring the links between software development task type, team attitudes and task completion performance: Insights from the Jazz repository. Information and Software Technology, 97, 10-25

[32] Meyer, A. N., Fritz, T., Murphy, G. C., & Zimmermann, T. (2014, November). Software developers' perceptions of productivity. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (pp. 19-29). ACM

[33] Graziotin, D., Wang, X., Abrahamsson, P., 2014b. Happy software developers solve problems better: psychological measurements in empirical software engineering. PeerJ 2, e289.

[34] Cruzes, D. S., Moe, N. B., & Dybå, T. (2016, August). Communication between developers and testers in distributed continuous agile testing. In 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE) (pp. 59-68). IEEE.

[35] Destefanis, G., Ortu, M., Counsell, S., Swift, S., Marchesi, M., & Tonelli, R. (2016). Software development: do good manners matter?. PeerJ Computer Science, 2, e73.

[36] Mirhosseini, S., & Parnin, C. (2017, October). Can automated pull requests encourage software developers to upgrade out-of-date dependencies?. In Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering (pp. 84-94). IEEE Press.

[37] Licorish, S. A., & MacDonell, S. G. (2017). Exploring software developers' work practices: Task differences, participation, engagement, and speed of task resolution. Information & Management, 54(3), 364–382

[38] Graziotin, D., Fagerholm, F., Wang, X. & Abrahamsson, P. (2017, May) Unhappy developers: Bad for themselves, bad for process, and bad for software product. In Proceedings of the 39th International Conference on Software Engineering Companion (pp. 362-364) IEEE Press

[39] Graziotin, D., Fagerholm, F., Wang, X., & Abrahamsson, P. (2017, May). Consequences of unhappiness while developing software. In Proceedings of the 2nd International Workshop on Emotion Awareness in Software Engineering (pp. 42-47). IEEE Press.

[40] Wang, Y. Y., Davison, R. M., & Huang, Q. (2017) Understanding How Software Developers Innovate in an IT-Enabled Work Environment: A Role Perceptions Perspective

[41] Graziotin, D., Fagerholm, F., Wang, X., & Abrahamsson, P. (2018). What happens when software developers are (un) happy. Journal of Systems and Software, 140, 32-47.

[42] Brown, J. A., Ivanov, V., Rogers, A., Succi, G., Tormasov, A., & Yi, J. (2018). Toward a Better Understanding of How to Develop Software Under Stress-Drafting the Lines for Future Research. arXiv preprint arXiv:1804.09044.

[43] Rezvani, A., Barrett, R., & Khosravi, P. (2018). Investigating the relationships among team emotional intelligence, trust, conflict and team performance. Team Performance Management: An International Journal.

[44] Lenberg, P., & Feldt, R. (2018, May). Psychological safety and norm clarity in software engineering teams. In 2018 IEEE/ACM 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE) (pp. 79-86). IEE

[45] Meyer, A., Barr, E. T., Bird, C., & Zimmermann, T. (2019). Today was a Good Day: The Daily Life of Software Developers. IEEE Transactions on Software Engineering

[46] Rezvani, A., & Khosravi,P. (2019). Emotional intelligence: The key to mitigating stress and fostering trust among software developers working on information system projects. International Journal of Information Management, 48, 139-15

[47] Moe, N. B., Dahl, B., Stray, V., Karlsen, L. S., & Schjødt-Osmo, S. (2019, January). Team Autonomy in Large-Scale Agile In Proceedings of the 52nd Hawaii International Conference on System Sciences

[48] Esaulova,I., Semenova,I., & Molodchik,N.(2019,January). The Impact of the Organizational Culture on the Employee Proactivity: Empirical Study in Innovation Oriented Companies. In 2nd International Scientificconference on New Industrialization: Global, national, regional dimension (SICNI 2018). Atlantis Press.

[49] Jain, P., Sharma, A., & Ahuja, L. (2018). Software Maintainability Estimation in Agile Software Development. International Journal of Open Source Software and Processes (IJOSSP), 9(4), 65-78.

[50] Tahir, M., Khan, F., Babar M., Arif, F., & Khan, F. (2016). Framework for better reusability in component based software engineering. the Journal of Applied Environmental and Biological Sciences (JAEBS), 6(4S), 77-81.

[51] Yadav, H. B., & Yadav, D. K. (2017). Early software

reliability analysis using reliability relevant software metrics. International Journal of System Assurance Engineering and Management, 8(4), 2097-2108.

[52] Rafi, U., Mustafa, T., Iqbal, N., & Zafar, W. (2015). US-Scrum: A Methodology for Developing Software with Enhanced Correctness, Usability and Security. International Journal of Scientific & Engineering Research, 6(9), 377.

[53] Arayici, Y., Fernando, T., Munoz, V., & Bassanino, M. (2018). Interoperability specification development for integrated BIM use in performance based design. Automation in Construction, 85, 167-181.

[54] Harrell, S. L., Kitson, J., Bird, R., Pennycook, S. J., Sewall, J., Jacobsen, D., ... & Robey, R. (2018, November). Effective performance portability. In 2018 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC) (pp. 24-36). IEEE.

[55] Huda, M., Arya, Y. D. S., & Khan, M. H. (2015). Testability quantification framework of object oriented software: a new perspective. International Journal of Advanced Research in Computer and Communication Engineering, 4(1), 286.

[56] Spriestersbach, A., & Springer, T. (2004, April). Quality attributes in mobile web application development. In International Conference on Product Focused Software Process Improvement (pp. 120-130). Springer, Berlin, Heidelberg

[57] Ashraf, M. Usman, et al. "Performance and power efficient massive parallel computational model for HPC heterogeneous exascale systems." IEEE Access 6 (2018): 23095-23107.

[58] Ashraf, Muhammad Usman, Rida Qayyum, and Hina Ejaz. "STATE-OF-THE-ART, CHALLENGES: PRIVACY PROVISIONING IN TTP LOCATION BASED SERVICES SYSTEMS." International Journal of Advanced Research in Computer Science (IJARCS) 10.2 (2019).

[59] Fatima, Fariha, Saqib Ali, and Muhammad Usman Ashraf. "Risk Reduction Activities Identification in Software Component Integration for Component Based Software Development (CBSD)." International Journal of Modern Education and Computer Science 9.4 (2017): 19.

[60] Siddiqui, Nasir, et al. "A highly nonlinear substitution-box (S-box) design using action of modular group on a projective line over a finite field." Plos one 15.11 (2020): e0241890.

[61] Javed, Rushba, et al. "Prediction and monitoring agents using weblogs for improved disaster recovery in cloud." Int. J. Inf. Technol. Comput. Sci.(IJITCS) 11.4 (2019): 9-17.

[62] Ashraf, Muhammad Usman, et al. "Provisioning quality of service for multimedia applications in cloud computing." Int. J. Inf. Technol. Comput. Sci.(IJITCS) 10.5 (2018): 40-47.

[63] Tariq, Saman, et al. "Measuring the Impact of Scope Changes on Project Plan Using EVM." IEEE Access 8 (2020): 154589-154613.

[64] Alsubhi, Khalid, M. Usman Ashraf, and Iqra Ilyas. "HBLP: A Privacy Protection Framework for TIP Attributes in NTTP-Based LBS Systems." IEEE Access 8 (2020): 67718-67734.

[65] Alrahhal, Mohamad Shady, et al. "AES-route server model for location based services in road networks." International Journal Of Advanced Computer Science And Applications 8.8 (2017): 361-368.

[66] Riaz, Shamsa, M. Usman Ashraf, and Ahmer Siddiq. "A Comparative Study of Big Data Tools and Deployment Platforms." 2020 International Conference on Engineering and Emerging Technologies (ICEET). IEEE, 2020.

[67] Alsubhi, Khalid, et al. "MEACC: an energy-efficient framework for smart devices using cloud computing systems." Frontiers of Information Technology & Electronic Engineering 21.6 (2020): 917-930.

[68] Abid, Usra, and M. Usman Ashraf. "A Critical Survey On Privacy Prevelling In Collaborative Filtring Recomender System: Challenges, State-Of-The-Art Methods And Future Directions." 2020 International Conference on Engineering and Emerging Technologies (ICEET). IEEE, 2020.

[69] Manzoor, Anam, et al. "Inferring Emotion Tags from Object Images Using Convolutional Neural Network." Applied Sciences 10.15 (2020): 5333.

[70] Alsubhi, K., et al. "A Tool for Translating sequential source code to parallel code written in C++ and OpenACC." 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA). IEEE, 2019.