

# Research Trends in Improving the Quality of Software Products with a Focus on Team Characteristics

Hamidreza Asfa, and \*Taghi Javdani Gandomani, *Senior Member, IEEE*

**Abstract**— Many factors play a role in creating a software product in the software production process. Quality measurement criteria have always been focused on the final product and output of the programming team in the world of software engineering. The emphasis of studies in the field of software quality is mainly on product features or system development stages. In addition to reviewing the related models, this study seeks to enter the workspace of software development teams by changing the perspective from the product to the developers and examines the effect of team characteristics on product quality, research ambiguities, and challenges that software researchers may encounter. Some of the most important issues that can be mentioned as the research gaps include many challenges and issues, including the lack of models with different attitudes toward the product, lack of field studies, and insufficient attention to team characteristics.

**Index Terms**— *Software Quality, Team Features, Software Development, Team Members' Satisfaction, Teamwork Quality.*

## I. INTRODUCTION

Software production processes have become particularly important to meet the needs of organizations and industries at different levels of work and information in today's advanced industrial world, which we are witnessing the growth, innovation, and development of new products and services. These processes involve a set of activities, each of which plays a vital role in the development of the right product. Some of the factors that play a serious role in the software production process include: collecting accurate and complete information, analysis of collected data correctly, creating a prototype based on customer demand, having a skilled programmer team, supporting and providing suitable software packages, etc. So far, various standards and criteria have been introduced to define and present good software based on which the final product and its quality are measured. In addition to providing new versions of the software, it is necessary to provide appropriate support along with the product to the customer to have a suitable and up-to-date product. The software development team meets the customer's needs according to the customer's needs and desires by increasing product efficiency and offering new methods, and

ensures the quality of its product. The software development team consists of several people, each of whom has unique characteristics such as specific personality type, emotions, and feelings, intellectual and psychological culture, level of literacy and education, experience, etc. These characteristics can have a significant effect on the efficiency of the person and teamwork. There is no standard work procedure that can be used for all software developers in software teams, and the requirements of the project and product must be considered in its implementation. The importance of manpower increases in such circumstances, and in fact, the whole project is considered a kind of human resource management, and the need for proper management of this resource is strongly felt. Software is considered a product of teamwork that shows the skill and ability of its developer team. If team members are selected in a more informed way, the end product will be more efficient and will be closer to the customer demand. A software product is a measure by which the ability and skill of developers are measured, and not only the management and proper knowledge of the team can be effective on tasks, but many issues are affected by it. To know, manage and assess the team properly, the main focus should be on the people who are the main capital of the project. It is very important to know and recognize the characteristics such as the level of skill, experience, education, and personal characteristics that will affect the efficiency of the developer team and the final product. The concept of quality in software is summarized in Section 2 of this study. Then, the software quality models are reviewed in Section 3. Studies focusing on team characteristics in software products are reviewed in Section 4. Open-ended questions on the quality of software products with a focus on team characteristics are addressed in Section 5. Then, a discussion is presented in section 6, and finally, section 7 presents the general conclusion.

## II. THE CONCEPT OF QUALITY IN SOFTWARE

The quality of software products is very important in software engineering because, in addition to customer satisfaction, it will be accompanied by the satisfaction of the development team and will lead to the creation and continuation of motivation, creativity, and innovation for the manufacturer. There are many definitions of quality. Among all the definitions proposed, the definition provided by the Institute of Electrical and Electronics Engineers (IEEE) [1] in

Hr. Asfa is with Department of Software Engineering, Isfahan Branch, Islamic Azad University, Iran. (email: hr.asfa@gmail.com)

\*T. Javdani Gandomani is with Department of Computer Science, Shahrekord University, Iran (e-mail: javdani@sku.ac.ir, javdani@ieee.org)

terms of quality conveys well the concept of quality: "The degree to which a system, component, or process meets specified requirements. ... "However, quality depends on the extent to which the identified needs correctly reflect the needs, interests, and expectations of individuals." Software Quality Assurance (SQA) is another important point in software quality, which IEEE has also provided for its acceptable definition: "Software quality assurance (SQA) is a means and practice of monitoring the software engineering processes and methods used in a project to ensure proper quality of the software." A concept was defined for the developed software quality assurance due to the importance of software quality and the important impact of planning and budgeting on the quality of the final product: "A set of activities that define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate and produce software products of suitable quality for their intended purposes (in accordance with scheduling and budgeting requirements) [2]-[4]. According to the above definition, the concept of software quality refers to the practical and managerial aspects of software production, development, and maintenance, and one of its main goals is also to reduce costs and increase productivity.

### III. AN OVERVIEW OF SOFTWARE QUALITY MODELS

Many studies have been done on software quality, in which important factors are addressed. In each study. Some important models in software engineering are discussed in this section:

#### A. McCall Software Quality Model

McCall [5] first proposed a model to determine and examine the relationship between quality factors and product evaluation criteria. Three views were expressed in this model for grouping features in the software production process, which are:

- Operation perspective: This perspective refers to the qualitative factors that indicate the extent to which the specifications and features of the software are met.
- Revision perspective: This perspective defines qualitative factors that affect the ability to change the software product.
- Transition perspective: Qualitative factors affecting the ability to change the software product.

#### B. Boehm model

In 1978, B.W. Boehm introduced his software quality model. B.W. Boehm, in a quantitative evaluation of software quality, added a series of features to the McCall model with an emphasis on software maintenance. He defined a hierarchical model of quality characteristics and tried to define software quality as a set of characteristics and criteria [6]. According to the Boehm model, three basic requirements including functionality, usability, maintainability, and portability is defined for software, each of which is divided into

subcategories. Also, this model added software evaluation-related considerations due to its type of application and hardware-related features.

#### C. Functionality, Usability, Reliability, Performance, Supportability (FURPS) model

Grady et al. presented a model called FURPS that divides software requirements into two groups: operational requirements and non-operational requirements.

- Operational requirements are defined with the required input and output.
- Non-operational requirements include five features: functionality, usability, reliability, efficiency, and supportability.

#### D. ISO/IEC 9126 model

The ISO/IEC 9126 model was published by the International Organization for Standardization (ISO) due to the software industry's urgent need to standardize software evaluation and was revised and completed by ISO experts in 2001. This model was presented in two levels which the quality of the software product is divided into six main quality characteristics in the first level of the model, each of which consists of several sub-characteristics. The relationship between the features of the first level of the model with the 21 sub-features of the model with the second level is one-to-many, so there is the least overlap in this model [7]-[10].

#### E. Dromey, R. G model

Dromey R. G (1995), during a study on a model for software product quality, concludes that significant gains in software quality are not achieved until a comprehensive model of software product quality is available. Dromey, R. G believes that the ISO/IEC 9126 standard, which is known as the Software Product Evaluation Characteristics Standard, is considered a high-level framework to describe the quality of the software product. The biggest challenge in proposing any software model is finding a framework that can make it flexible and intellectually manageable in a constructive way. Identifying the high-level quality attributes of a small set and the generality of this top-down approach for subsets is a common approach for developing a software model [11].

Software evaluations are different and require more dynamic issues to model processes. According to him, the philosophy of this model is to be able to include a wide range of systems with different applications. Therefore, they presented a model that can be summarized in five steps as follows:

- Select a set of high-level quality attributes needed for evaluation
- Preparing a list of system components
- Identifying quality attributes for each component of the system using the components of the previous step

- Making the decision on how each feature affects quality attributes
- Model evaluation

TABLE I  
INTRODUCTION OF MODELS, ADVANTAGES, AND DISADVANTAGES

Model Name	Summary of Work Done	Advantages	Disadvantages
McCall's Model	Three perspectives in the software production process: functional, revision, transformation	Determining effective factors in the software production process	Lack of implementation in different projects
Boehm Model	Adding new features to McCall's model	1- Useful and practical program, portability, maintenance 2- Improving McCall's model	Not paying attention to the characteristics of the programmer team
FURPS Model	Classification of software quality features	Presentation of FURPS model	Not paying attention to the characteristics of the programmer team
ISO-9126 model	Standardization of software production and evaluation process	Separation of internal and external quality features of a software	Lack of complete overlap between first and second-level features
Dromey Model	Emphasis on the existence of a comprehensive model of software product quality	Use the standard ISO-9126 for the evaluation of the soft drive product	Not paying attention to the characteristics of the programmer team

#### IV. WORKS RELATED TO TEAM CHARACTERISTICS IN SOFTWARE PRODUCTS

Research has also been done in this field that examines quality through criteria such as the personality of individuals in software teams. A review of the literature shows that several studies have been carried out on the relationship between software product quality and team structure, and the personality traits of developers.

Acuña [12] during a study entitled "The relationship between personality, team process, and job characteristics with job satisfaction and software quality, concluded that teams that scored higher on personality factors such as conscientiousness were more satisfied with their job. He also pointed out the nature of extraversion in the team and its effective role in the high interaction between team members.

Soomro Arjumand Bano [13], during a systematic literature review, examined the effect of people's personalities on the efficiency of an engineering team. For their research, they used the methods of Kitchenham & Charters [14] and Kitchenham & Brereton [15] and used a Meta-summary method in which qualitative findings are combined with quantitative findings to combine their findings instead of statistical summaries of data.

Cruz Shirley [16] Carried out a study on the role of personality in software engineering aimed to provide a context for studies and experiences and sought to provide a precise

definition of human personality which was defined by Ryckman as "dynamic and organized". In this study, nine different issues were examined according to the effect of individual personality in software engineering, including pair programming, Team Effectiveness, Individual Performance, Software Process Allocation, Behavior and Preference, Education, Project Manager Effectiveness, Personality Test Application, and job retention.

It should be noted that, each of the individual characters mentioned has also been examined separately in some studies; for example, some researchers examined the influence of personality on pair programming [17]-[19]. Also, some studies have examined the relationship between individual talent and personality traits in successful programming [20], [21].

Meanwhile, some researchers have paid more attention to personality. For example, McDonald and Edwards used personality tests and their application in software teams [22]. On the other hand, some researchers examined the relationship between personality and individual performance of team members [23], [24].

According to research conducted by [12], team performance cannot be due to the combination of team personality and the task assigned to them but also depends on the interactive effects of team behavior. In general, in the team performance literature, the team behavioral model is mainly based on the McGrath model called McGrath's Input-Process-Output Model [25], [26].

Some researchers also examined extraversion, conscientiousness, and problem-solving performance [27], [28]. Yang and Tang [29] examined social dependence and incompatibility in team processes and software systems development performance.

So far, research has been done in the field of agile software teams and knowledge management, and to some extent, these challenges have been addressed in these teams but there is a need for further study and research [30]-[37].

#### V. DISCUSSION

According to [16], no evidence has been presented so far to show that there is a significant relationship between the personality type of team members with the talent and success of programming. In recent years, researchers have tried to identify the relationship between personality and performance. These studies have been carried out based on five dimensions or personality factors, including anger, extraversion, openness to experience<sup>1</sup>, adaptation, and conscientiousness [38]-[41]. Also, there is no consensus on the definition of human personality, although personality typically refers to individual differences [13], [16], [42].

Oakin model is one of the most well-known personality models that has been used in many studies to understand the impact of personality traits on the teamwork environment [43]. Furthermore, the Myers-Briggs Type Indicator (MBTI) is

<sup>1</sup> Openness to experience, or simply openness, is a basic personality trait denoting receptivity to new ideas and new experiences.

another questionnaire that is widely used in software engineering n personality-related research [44], [45].

As mentioned before, and according to the results of studies conducted in the field of software quality and team features, it can be concluded that in the models' section, the majority of these models pay attention to the product features and less to the manufacturer team or some research focus on the team characteristics and some parameters are examined individually.

Also, one of the challenges in this field is that other previous articles and studies have been used by many existing studies, especially in models, and few field studies have been done. Many studies have limitations due to the structures and communications of software team members, which are sometimes unavoidable and sometimes related to how the research team works.

It seems that focusing on team features is considered one of the ways to achieve the desired quality in software products. Of course, there are a variety of research questions and trends in these challenges. Some of the most important ambiguities and challenges are presented in the next section.

#### IV. NEW RESEARCH TRENDS

As mentioned in the previous section, there are still ambiguities, and challenges in the field of software quality, which can be provided to researchers for further research, and attention to them can have a positive impact on improving the quality of software products, the most important items include:

- Why have fewer field studies been used in this field? McCall, for the first time, discussed the quality attributes of software, and this field was raised in the software industry following his ideation. In the software industry, researchers began to research and study in this field. Given that these studies are limited to the articles and writings written by previous people, and fewer studies have been done in the field, so it seems necessary to conduct field research and studies.
- What characteristics should a team have to be able to provide customer satisfaction in addition to producing the right products? Software teams are composed of people with various characteristics; the result of the cooperation of these teams is manifested in the form of a software product. Answering this question makes it easier to respond to customer requests. An issue that may have received less attention.
- Is it possible to provide a model based on team members' characteristics that improve the software product's quality? Different models have studied the quality of previous years by introducing various quality factors, but in all these models, the output determines the skill level of a software team. Therefore, it can be expected that a new model can be presented by focusing on the people working in the team and using their characteristics. According to the above

questions, it can be concluded that there are still various aspects and concepts in the field of software quality that need to be further researched and studied. It can lead us to have higher quality software products by focusing on the individual characteristics of the teams and answering each question. According to the stated content, the figure (1) is a proposed model of the role of the enabling factors in the production of software products.

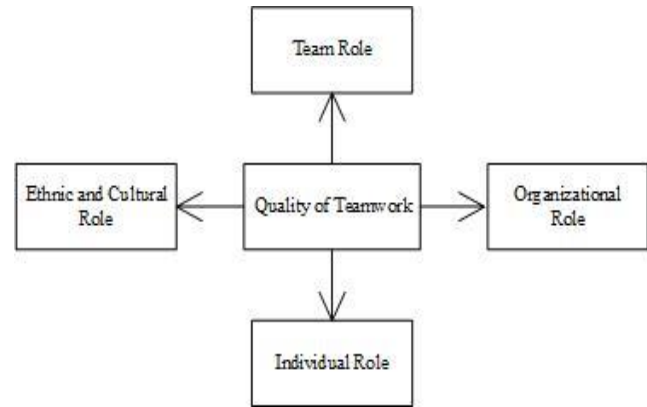


Fig 1. The proposed model based on the role of teamwork

#### IV. CONCLUSION

Software manufacturing companies are always looking for effective steps to attract their customers, which is made possible by providing up-to-date, simple, efficient, and appropriate products. In the meantime, efforts have been made based on the production of software that has good features, quality, and efficiency, and also standards and policies have been developed for it. But the issue of the team that produces this product has received less attention. Reviewing the literature, it seems that everyone has paid attention to one aspect of this research in previous studies, but it was not complete, so it seemed appropriate to examine it from a complete perspective. Several questions are raised about the quality of software products that pay attention to them, especially in the field of team characteristics, as the main purpose of this study can be one of the proposed research areas in this field.

#### References

- [1] D. Galin, *Software quality: concepts and practice*: John Wiley & Sons, 2018.
- [2] S. Goericke, *The future of software quality assurance*: Springer Nature, 2020.
- [3] P. Jain, A. Sharma, and L. Ahuja, "A customized quality model for software quality assurance in agile environment," *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 14, no. 3, pp. 64-77, 2019.
- [4] P. Temple, M. Acher, G. Perrouin, B. Biggio, J.-M. Jézéquel, and F. Roli, "Towards quality assurance of software product lines with adversarial configurations." pp. 277-288.
- [5] J. A. McCall, P. K. Richards, and G. F. Walters, "Factors in software quality: preliminary handbook on software quality for an acquisition manager," *General Electric*, vol. 1, pp. 3, 1977.

- [6] B. W. Boehm, J. R. Brown, and M. Lipow, "Quantitative evaluation of software quality." pp. 592-605.
- [7] A. Abran, "Software engineering-Product quality-Part 3: Internal metrics," *ISO draft for Review*, 2000.
- [8] F. Coallier, "Software engineering-Product quality-Part 1: Quality model," *International Organization for Standardization: Geneva, Switzerland*, 2001.
- [9] I. ISO, and T. IEC, "9126-2: Software Engineering-Product Quality-Part 2: External Metrics," *International Organization for Standardization, Geneva, Switzerland*, 2003.
- [10] I. ISO, and T. IEC, "9126-4:-Software Engineering-Product Quality-Part 4:-Quality in Use Metrics," *International Organization for Standardization, Geneva, Switzerland*, 2004.
- [11] R. G. Dromey, "A model for software product quality," *IEEE Transactions on Software Engineering*, vol. 21, no. 2, pp. 146-162, 1995.
- [12] S. T. Acuña, M. Gómez, and N. Juristo, "How do personality, team processes and task characteristics relate to job satisfaction and software quality?," *Information and Software Technology*, vol. 51, no. 3, pp. 627-639, 2009.
- [13] A. B. Soomro, N. Salleh, E. Mendes, J. Grundy, G. Burch, and A. Nordin, "The effect of software engineers' personality traits on team climate and performance: A Systematic Literature Review," *Information and Software Technology*, vol. 73, pp. 52-65, 2016.
- [14] B. Kitchenham, and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.
- [15] B. Kitchenham, and P. Brereton, "A systematic review of systematic review process research in software engineering," *Information and Software Technology*, vol. 55, no. 12, pp. 2049-2075, 2013.
- [16] S. Cruz, F. Q. da Silva, and L. F. Capretz, "Forty years of research on personality in software engineering: A mapping study," *Computers in Human Behavior*, vol. 46, pp. 94-113, 2015.
- [17] Hannay J. E, Arisholm E, Engvik H, and D. I. K. Sjoberg, "Effects of personality on pair programming," *IEEE Transactions on Software Engineering*, 36(1), pp. 61-80, 2010.
- [18] N. Salleh, E. Mendes, J. Grundy, and G. S. J. Burch, "An empirical study of the effects of personality in pair programming using the five-factor model." pp. 214-225.
- [19] B. Xu, S. Yan, K. Gao, Y. Zhang, and G. Yu, "Influence of periodic role switching intervals on pair programming effectiveness." pp. 3-14.
- [20] K. E. Pocius, "Personality factors in human-computer interaction: A review of the literature," *Computers in Human Behavior*, vol. 7, no. 3, pp. 103-135, 1991.
- [21] G. M. Weinberg, *The psychology of computer programming*: Van Nostrand Reinhold New York, 1971.
- [22] B. S. Mcdonald, and H. M. Edwards, "Who should test whom?," *Communications of the ACM*, 50(1), pp. 66-71, 2007.
- [23] M. R. Barrick, and M. K. Mount, "The big five personality dimensions and job performance: a meta-analysis," *Personnel Psychology*, vol. 44, no. 1, pp. 1-26, 1991.
- [24] N. Qamar, and A. A. Malik, "Birds of a Feather Gel Together: Impact of Team Homogeneity on Software Quality and Team Productivity," *IEEE Access*, vol. 7, pp. 96827-96840, 2019.
- [25] J. E. McGrath, *Groups: Interaction and performance*: Prentice-Hall Englewood Cliffs, NJ, 1984.
- [26] A. C. D. Batista, R. M. de Souza, F. Q. da Silva, L. de Almeida Melo, and G. Marsicano, "Teamwork quality and team success in software development: A non-exact replication study." pp. 1-11.
- [27] B. Barry, and G. L. Stewart, "Composition, process, and performance in self-managed groups: The role of personality," *Journal of Applied Psychology*, vol. 82, no. 1, pp. 62, 1997.
- [28] K. N. Javaras, M. Williams, and A. R. Baskin-Sommers, "Psychological interventions potentially useful for increasing conscientiousness," *Personality Disorders: Theory, Research, and Treatment*, vol. 10, no. 1, pp. 13, 2019.
- [29] H.-L. Yang, and J.-H. Tang, "Team structure and team performance in IS development: a social network perspective," *Information & Management*, vol. 41, no. 3, pp. 335-349, 2004.
- [30] S. Ghobadi, and L. Mathiassen, "Perceived barriers to effective knowledge sharing in agile software teams," *Information Systems Journal*, vol. 26, no. 2, pp. 95-125, 2016.
- [31] D. Šmite, N. B. Moe, A. Šāblis, and C. Wohlin, "Software teams and their knowledge networks in large-scale software development," *Information and Software Technology*, vol. 86, pp. 71-86, 2017.
- [32] C. Khalil, and S. Khalil, "Exploring knowledge management in agile software development organizations," *International Entrepreneurship and Management Journal*, vol. 16, no. 2, pp. 555-569, 2020.
- [33] N. Amiri, and T. J. Gandomani, "Employing Mark Knowledge Management Model to Improve Agile Team Productivity: An Extension of the Scrum Process," *Majlesi Journal of Mechatronic Systems*, vol. 9, no. 2, pp. 29-35, 2020.
- [34] T. Javdani Gandomani, Z. Tavakoli, M. Ziaei Nafchi, and M. N. Sarpiri, "Adapting Scrum Process with 7C Knowledge Management Model." pp. 056-059.
- [35] G. Marsicano, F. Q. da Silva, C. B. Seaman, and B. G. Adaid-Castro, "The Teamwork Process Antecedents (TPA) questionnaire: developing and validating a comprehensive measure for assessing antecedents of teamwork process quality," *Empirical Software Engineering*, vol. 25, no. 5, pp. 3928-3976, 2020.
- [36] A. Radhakrishnan, J. Zaveri, D. David, and J. S. Davis, "The impact of project team characteristics and client collaboration on project agility and project success: An empirical study," *European Management Journal*, 2021.
- [37] R. Subramaniam, S. Nakkeeran, and S. Mohapatra, "Team Work Quality," *Team Work Quality*, pp. 11-12: Emerald Publishing Limited, 2021.
- [38] A. English, R. L. Griffith, and L. A. Steelman, "Team performance: The effect of team conscientiousness and task type," *Small Group Research*, vol. 35, no. 6, pp. 643-665, 2004.
- [39] G. A. Neuman, and J. Wright, "Team effectiveness: beyond skills and cognitive ability," *Journal of Applied Psychology*, vol. 84, no. 3, pp. 376, 1999.
- [40] I. Sommerville, and T. Rodden, "Human, social and organisational influences on the software process," *Software Process*, vol. 4, pp. 89-100, 1996.
- [41] A. E. Van Vianen, and C. K. De Dreu, "Personality in teams: Its relationship to social cohesion, task cohesion, and team performance," *European Journal of Work and Organizational Psychology*, vol. 10, no. 2, pp. 97-120, 2001.
- [42] S. S. Cruz, F. Q. da Silva, C. V. Monteiro, P. Santos, and I. Rossilei, "Personality in software engineering: Preliminary findings from a systematic literature review." pp. 1-10.
- [43] S. L. Kichuk, and W. H. Wiesner, "The big five personality factors and team performance: implications for selecting successful product design teams," *Journal of Engineering and Technology management*, vol. 14, no. 3-4, pp. 195-221, 1997.
- [44] A. B. Cunha, A. G. Canen, and M. A. Capretz, "Personalities, cultures and software modeling: Questions, scenarios and research directions." pp. 23-31.
- [45] A. Furnham, "Myers-Briggs type indicator (MBTI)," *Encyclopedia of personality and individual differences*, pp. 3059-3062, 2020.