

## Research Article

# Multicriteria Decision Making to Continuous Software Improvement Based on Quality Management, Assurance, and Metrics

Lei Sun <sup>1</sup>, Shah Nazir <sup>2</sup>, and Anwar Hussain <sup>2</sup>

<sup>1</sup>School of Information and Electrical Engineering, Hebei University of Engineering, HanDan 056038, China

<sup>2</sup>Department of Computer Science, University of Swabi, Swabi, Pakistan

Correspondence should be addressed to Lei Sun; [sl65536@sina.com](mailto:sl65536@sina.com) and Shah Nazir; [snsahnzr@gmail.com](mailto:snsahnzr@gmail.com)

Received 10 March 2021; Revised 5 April 2021; Accepted 12 April 2021; Published 21 April 2021

Academic Editor: Sikandar Ali

Copyright © 2021 Lei Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software process improvement methodologies support to incessantly regulate and advance the process of software for its performance development. This development arises in terms of software product quality, time, changes reduction, and so on. The aim of software process improvement is to make software process effective and to increase quality of software product through continuous evaluation. Various approaches and frameworks are in practice for software process improvement. Early decision making concerning continuous software improvement can ultimately lead to successful software developments and growing software industry and business of software. Keeping in view the early decision making regarding continuous software improvement, the proposed study has considered a decision support system (DSS) for making decision based on multicriteria against quality management, assurance, and metrics. The necessary factors and alternatives for continuous software improvement were considered. The process of the proposed study was experimentally conducted, and results were good enough for supporting the study. The study will help practitioners in early decision making regarding software improvement based on quality management.

## 1. Introduction

Software process improvement approaches support to continually adjust and improve the process of software for its performance improvement. This improvement occurs in terms of software product quality, time, change reduction, and so on. The aim of software process improvement is to make the software process further effective and to increase quality of software product through continuous evaluation. Various approaches and frameworks are in practice for software process improvement. Lean software development disseminates to support improvement of the process through two significant principles which include identification of process waste and consideration of interactions in the software process. Software process improvement through the Lean Measurement was presented. The approach is based on the paradigm of quality improvement and practices of lean software development [1]. The approach

permits to evaluate the development process evaluation and taking continuous actions for arriving at a more lean software process over time. Reliability of software is considered as one of the most significant sides of software quality. Most of the available approaches are focusing on testing of stems and some other stages of the development life cycle. However, they are not efficient in improvement of reliability and initial identification of problem. For overcoming these issues, research work was established for predictive linkages between other entities which can measure and control primarily in the development cycle, software reliability, and through such prediction relations to drive continuous reliability improvement [2].

For process improvement, staged models have been efficiently considered by the organization of software. The associations between these models and organization-wide continuous process improvement have not been elaborated comprehensively. Research work has been presented for

analyzing activities of software process improvement and perceptions of employees in two diverse software companies based on certification of CMMI level 3. Two phases of multiple case studies were performed for analysis. Primarily, process improvement suggestions submitted by the employees are examined according to three aspects, such as submitter, timing, and content characteristics. After this, the perception of employees concerning the organization's varied activities of continuous process improvement is examined by a factor analysis questionnaire of the employees. Results of the study have shown that there are deficiencies concerning organization-wide continuous process improvement. It is observed that activities of software process improvement are not executed in a continuous way [3]. Global software development plays a significant role in software engineering and development of software. Various challenges are faced to global software development concerning software process improvement. Prioritization-based taxonomy of factors for software process improvement are developed through approach of the fuzzy analytic hierarchy process. Twenty-one features were taken out from the existing research available in the literature and were further assessed by survey of the questionnaire with experts of software process improvement. The fuzzy AHP which is a multicriteria decision-making approach was considered for prioritization and identifying features along with their categories. The approach can work in situations of uncertainty and vagueness in process improvement experts' opinion [4].

Software companies required to incessantly enhance the process of their services. For satisfaction of business goals with greater effectiveness and satisfaction of customer, the France telecommunication research and development designed a software process improvement programme and implementation of risk management in software process improvement in 1999 [5]. Kanoun [6] has emphasized on the collection of data and program analysis for reliability of software improvements. The study has presented the data collection program objectives, presented some achievements and successful stories associated to software reliability improvements, and discussed data collection practice aspects, validation, and processing for effective programs analysis and collection of data. The successful stories have described the improvement in production and reliability is achieved nearly with no additional cost and at reduced time. For the purpose of reliability, the study has considered three key activities including trend analysis, descriptive analysis, and reliability assessment. The experimental results of the study are based on experience in processing failure data gathered on real-life software. Properties of software such as credibility and resilience were considered for optimum selection of software design. The study has used an analytic hierarchy process which is a multicriteria decision-making approach [7]. The selection of the software component is considered difficult due to nonavailability of complete information of the software components. Fuzzy logic is considered as a tool of overcoming the issue of uncertainty and vagueness for selection of software components which enhanced work for software process improvement [8].

The current study has adopted a decision support system for making early decision based on multicriteria against quality management, assurance, and metrics. The essential factors and alternatives for continuous software improvement were considered. The process of the study was experimentally conducted, and results were good enough for supporting the study. The study will help practitioners in early decision making regarding software improvement based on quality management.

This paper is outlined as follows: Section 2 briefly describes the related work to the proposed study. Section 3 is the representation of software process improvement and details of the existing approaches available in literature. Section 4 briefly shows the decision support system to continuous process improvement. This paper is concluded in Section 5.

## 2. Related Work

Elliott et al. [9] have presented a cost-effective and evolutionary approach for implementing a software quality system requiring minimum resource and slight disruption for programme delivery. The proposed study has gained 40% improvement in the implementation of AWEplc software quality management over a period of 5 years. For enhancing the quality of software, a continuous improvement in the software process is the awful need of software development. Fukushima and Yamada [10] have discussed the activities of process improvement through monitoring and design quality and test quality assessed through quality assurance division. Moreover, the study examined these activity results quantitatively and discussed the consequences of the activities. The activities of process monitoring review management of the project, perceiving the solution and detection of quality, delivery, and cost-associated problems and enhance process management for leading a project for its accomplishment. Assessing activities of test quality and design quality quantifies the requirement specification completeness, test cases, and design specifications and enhances the process of software development for eradicating faults of software. Saputri and Lee [11] provided a study to support software practitioners for taking sustainability into account through providing organized guidelines for the process of software engineering. The research has proposed a framework presenting a metamodel for decomposing sustainability requirements and an evaluation approach for assessing sustainability achievements. The approach presented a framework which has integrated a scenario-based approach, a goal-based approach, and feature modeling for gathering sustainability-associated requirements and equivalent features. According to sustainability metrics and criteria, the sustainability evaluation was performed through machine learning approaches and software analysis for examining software products. The study was evaluated through an empirical study with academicians' and industry participants. The results revealed the suggested framework enhances ability of participants to consider aspects of sustainability in their task of software engineering with focus on requirements, design, and assessment.

Research in the area of decision making in Agile development is poor to understand and critical. Drury et al. [12] have analyzed decision making across four phases of the iteration cycle. These include iteration execution, iteration planning, iteration review, and iteration retrospective. A mixed method approach was considered where a focus group was conducted with 43 Agile developers and managers for determining decisions prepared at various points of the cycle. For considering as an example, six illustrative minicases were conducted for the obstacles traced in the focus group. Five diverse organizations were considered with 18 individuals in projects of Agile. These include a multinational communications company, a global consulting organization, two multinational software development companies, and a large museum organization. The research has contributed to present the literature of Agile software development through examining the decision made through an iteration cycle and traced six main obstacles to the decisions. The obstacles are conflicting priorities, unwillingness to commit to decisions, unstable resource availability, and lack of ownership, implementation, and empowerment. Effective practices of software engineering are required for strengthening small and medium enterprises. For the last few years, the community of software engineering has shown distinctive importance in software process improvement for increasing quality of the software product along with productivity of software development. Though, the trend shows that there is more focus on to stress that the realization of software process improvement is only conceivable for huge companies. Pino et al. [13] presented a comprehensive review of the case studies published on the efforts performed on the software process improvement. The aim of the study was to examine the available approaches on software process improvements focusing on small and medium enterprises with further reporting a case study. The research also has discussed the important issues associated to the area and to deliver an up-to-date research from which state-of-the-art activities of research can be assumed and planned.

### 3. Software Process Improvement

Börjesson et al. [14] presented an approach of dealing how agile enhancement practices can support organizations of software efficiently implementing a new process. An action research based enhancement inventiveness implementing a novel change request process and toll at the Swedish telecom company Ericsson AB in Gothenburg. The approach has recognized “guerilla tactic” for supporting agile enhancement practices and supporting effective employment. Gaston and Halloran [15] have suggested that, to achieve maximum advantages from process improvement program implementation, the organization has to become what is termed “a learning organization.” Software process evaluation “leads to the identification and selection of main activities for improvement and the continuous application of improvements to match business need.” The study has concluded the reasons by elaborating why substantial enhancements in performance were not gained. A significant importance is given to build an enterprise system to increase

revenues and cost optimization. Such systems are deployed through reusing the available systems more close with business needs. Mohan et al. [16] have provided a basis for achieving enhanced reliability with low defect density and high productivity along with affordability by a cost-efficient solution of customize software. The approach of the analytic network process has been considered for estimating the quality of service-affecting parameters. Zhang et al. [17] presented a study which has used a model of a nonhomogeneous Poisson process for assessing software reliability quantitatively for a nuclear power plant safety digital control system. The software test environment was recognized for analyzing the research object, and the research object was tested for obtaining software failure data. A software reliability model is incorporated with the data on nuclear power plants DCS software defects, for constructing a nonhomogeneous Poisson process model. Formulations including remaining failure, total failure, failure rate, time to next failure, and reliability were derived. The data obtained concerning software failure from the test were used for determining the variables of the model based on maximum likelihood estimation. The estimation was considered for quantitatively assessing reliability of software-associated indicators. The minimum correlation error approach was considered for analyzing the error through comparing the results of actual and mode prediction. The approach can assess the existing level of reliability of software and identify the influence of subsequent reliability test results. The results of the study have indicated to provide support for assessing the nuclear power plant DCS software reliability. Figure 1 represents the inventories in the software process [1]. The figure shows representation of various phases of the software process.

### 4. Decision Support System to Continuous Software Process Improvement

Venters et al. [18] have outlined generic perspectives and principles on sustainability with respect to software systems for providing a terminology and context to frame the discourse on architectures of software and sustainability. The study contributed in providing a roadmap and foundation of emerging research in the field of sustainable software architecture emphasizing latest trends and research challenges and issues. Colanzi et al. [19] have presented a systematic mapping study for four research questions. The study identified 134 papers published from 2009 to 2019. Also, 32 papers published related to the challenges track were summarized. Software developers are trying to deliver a great-quality software system. For addressing the issues associated to low-quality software, software process improvement models are used more for guiding the process of software development. Such a model includes capability maturity model integration. Implantation of such a model in a low software development organization is specifically challenging due to complexity and more cost. For addressing such issues, Chevers et al. [20] have proposed a simple framework of software process improvement for small organization to support software development. The framework

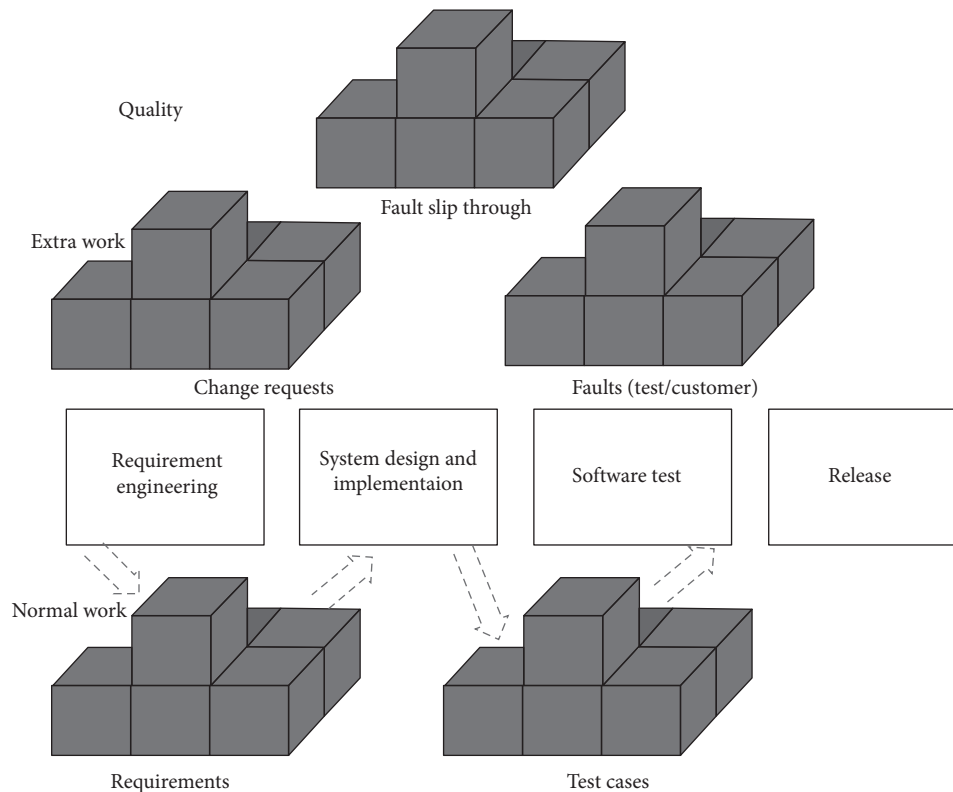


FIGURE 1: Inventories in the software process.

is based on capability maturity model integration. Cartaxo et al. [21] have proposed an approach to help researchers in transferring knowledge to practice of software engineering. The approach is considered based on the foundation of rapid reviews and evidence briefings. The study has considered how to apply the model along with the challenges hindering its adoption. Trienekens et al. [22] have presented a survey of software groups in a multinational organization. The study was considered by the steering committee of Philips software process improvement. This committee monitors the quality and status of software process improvement in global organization. The study discusses improvement drivers, improvement targets, and metrics and the level of recognition in software groups.

Various models have been proposed for analyzing the software process improvement. Some of the models are usable for very small entities. The research work has analyzed the dissimilarities from various economic regions and discusses that very small entities in developing countries normally can manage more staff due to much lower salaries in those countries [23]. The study then proposed a process improvement process as an addition to the existing ISO/IEC 29110. Software factors are those factors which influence the software. These can be generally categorized into two parts. The first type can be measured directly, for example, logical errors, and the second type can be measured indirectly, for example, maintainability. Nazir et al. [24] have presented an approach for the selection of components according to some quality criteria of ISO/IEC 25010:2011 [25]. The research work has adopted the analytic network process approach for

selection purpose. Shahzad et al. [26] have used the fuzzy ANP for software project assessment based on criteria defined in ISO/IEC 25010:2011. The proposed research has used the features of ISO/IEC 2382–36 for conducting the current study [27]. Figure 2 graphically depicts the hierarchy of criteria, alternatives, and goal of the proposed study.

The process of pairwise comparisons was carried out for the criteria, alternatives, and goal to achieve the optimum results associated to the best choice. Table 1 consists of the criteria for software project assessment and to consider software process improvement. These values are the derivations of expert opinion from different experts in the research domain.

The process of pairwise comparisons was carried out, and each row and column were normalized in order to get the eigen value for the criteria. Table 2 shows the normalized comparison obtained from Table 1.

The same process of pairwise comparison was carried out for alternatives in order to know about the best choice of available alternatives. Table 3 depicts the weights assigned to available alternatives.

Table 3 was normalized, and their pairwise comparison was made to obtain a normalized table. Table 4 represents the normalized table. In this table, the E.V. was found.

The same process of normalization and pairwise comparison were carried out for the rest of the criteria and alternatives. After performing pairwise comparison, all the E.V. of each tables are combined into a single table known as weighted super matrix. Table 5 is the representation of the weighted super matrix. This table is also called the summarized table of the normalization process.

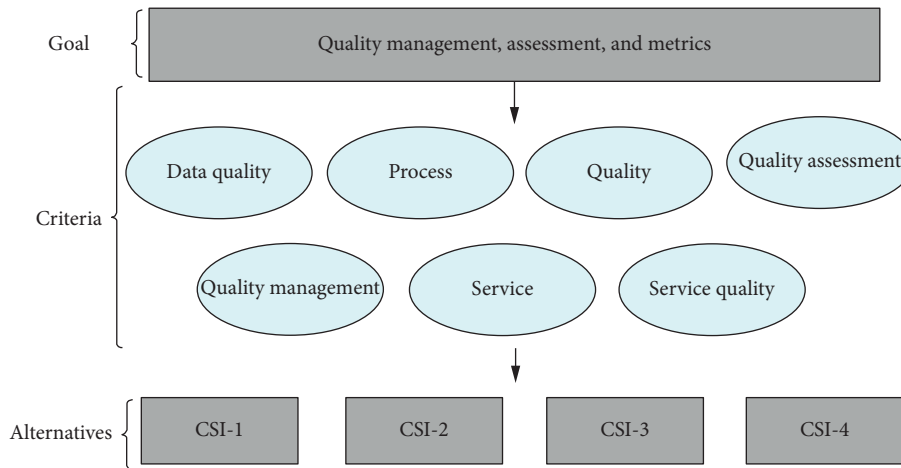


FIGURE 2: Hierarchy of goal, criteria, and alternatives.

TABLE 1: Criteria and their weights.

	Data quality	Process	Quality	Quality assessment	Quality management	Service	Service quality
Data quality	1	2	2	3	4	2	5
Process	1/2	1	2	3	2	3	3
Quality	1/2	1/2	1	4	3	3	2
Quality assessment	1/3	1/3	1/4	1	2	2	2
Quality management	1/4	1/2	1/3	1/2	1	2	3
Service	1/2	1/3	1/3	1/2	1/2	1	2
Service quality	1/5	1/2	1/2	1/2	1/3	1/2	1

TABLE 2: Normalization of criteria and their weights.

	Data quality	Process	Quality	Quality assessment	Quality management	Service	Service quality	E.V.
Data quality	0.30	0.39	0.31	0.24	0.31	0.148	0.278	0.283
Process	0.15	0.19	0.31	0.24	0.16	0.222	0.167	0.206
Quality	0.15	0.10	0.16	0.32	0.23	0.222	0.111	0.185
Quality assessment	0.10	0.06	0.04	0.08	0.16	0.148	0.111	0.100
Quality management	0.08	0.10	0.05	0.04	0.08	0.148	0.167	0.094
Service	0.15	0.06	0.05	0.04	0.04	0.074	0.111	0.076
Service quality	0.06	0.10	0.08	0.04	0.03	0.037	0.056	0.056

TABLE 3: Alternatives and their weights.

	CSI 1	CSI 2	CSI 3	CSI 4
CSI 1	1	3	2	4
CSI 2	1/3	1	2	3
CSI 3	1/2	1/2	1	2
CSI 4	1/4	1/3	1/2	1

TABLE 4: Normalization of alternatives and their weights.

	CSI 1	CSI 2	CSI 3	CSI 4	E.V.
CSI 1	0.48	0.62	0.36	0.40	0.47
CSI 2	0.16	0.21	0.36	0.30	0.26
CSI 3	0.24	0.10	0.18	0.20	0.18
CSI 4	0.12	0.07	0.09	0.10	0.09

The weighted super matrix was converted into a limit matrix to obtain the decision regarding the best choices of software process improvement. The limit matrix is

obtained by taking the power of the super matrix till all the values of each row become the same. Table 6 shows the limit matrix from which the decision can be made. This is the final table where decision regarding available alternatives can be made.

Figure 3 is the representation of final weights and ranks of available alternatives, that is, CSI 1, CSI 2, CSI 3, and CSI 4. From the figure, it can be seen that CSI 4 is the best option to be considered followed by CSI 3. This can further be used for any type of decision associated to continuous software process improvement.



TABLE 5: Weighted super matrix.

Node label		Features						CSPI				
		Data quality	Process	Quality	Quality assessment	Quality management	Service	Service quality	CSI 1	CSI 2	CSI 3	CSI 4
Use cases	Data quality	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.283	0.049	0.058	0.043
	Process	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.206	0.068	0.067	0.085
	Quality	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.185	0.102	0.113	0.086
	Quality assessment	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.100	0.132	0.119	0.123
	Quality management	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.094	0.151	0.182	0.157
	Service	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.076	0.214	0.213	0.224
	Service quality	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.056	0.283	0.247	0.282
CSPI	CSI 1	0.466	0.440	0.119	0.107	0.092	0.087	0.091	0.000	0.000	0.000	0.000
	CSI 2	0.258	0.285	0.176	0.171	0.161	0.196	0.179	0.000	0.000	0.000	0.000
	CSI 3	0.181	0.156	0.408	0.292	0.353	0.246	0.294	0.000	0.000	0.000	0.000
	CSI 4	0.095	0.119	0.297	0.430	0.394	0.472	0.435	0.000	0.000	0.000	0.000

TABLE 6: Limit matrix.

Node label		Features						CSPI				
		Data quality	Process	Quality	Quality assessment	Quality management	Service	Service quality	CSI 1	CSI 2	CSI 3	CSI 4
Use cases	Data quality	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.00	0.00	0.00	0.00
	Process	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.00	0.00	0.00	0.00
	Quality	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.00	0.00	0.00	0.00
	Quality assessment	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.00	0.00	0.00	0.00
	Quality management	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.00	0.00	0.00	0.00
	Service	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.00	0.00	0.00	0.00
	Service quality	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.00	0.00	0.00	0.00
CSPI	CSI 1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.16	0.16	0.16
	CSI 2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19	0.19	0.19	0.19
	CSI 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28	0.28	0.28	0.28
	CSI 4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.36	0.36	0.36	0.36

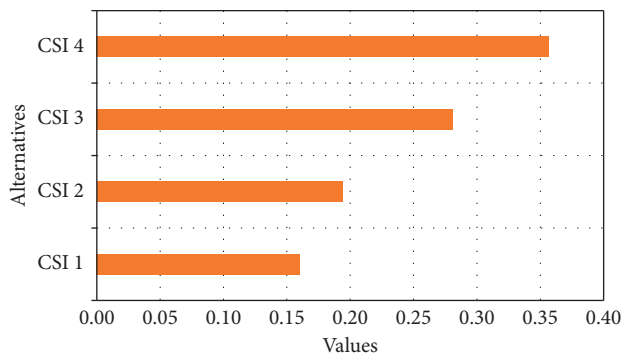


FIGURE 3: Ranking of available alternatives.

### 5. Conclusions

Software process improvement approaches help to continually improve the process of software for its routine improvement. This improvement arises in terms of software product time, quality, changes reduction, and so on.

Software process improvement aims to mark the software process more operative and to increase quality of the software product through continuous evaluation and management. Primary decision making regarding continuous software improvement can eventually lead to effective software developments and rising software industry. Keeping in view the early decision making regarding continuous software improvement, this study has proposed a decision support system for making decision based on multicriteria concerning quality management, assurance, and metrics. The required features and alternatives of continuous software improvement were considered. The process of the proposed study was experimentally carried out for continuous software improvement, and the results were good enough for supporting the study. The study will support experts in early decision making in relation to software improvement based on quality management. This study is an addition toward the success of software industry for continuous software improvement and business of software.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] K. Petersen and C. Wohlin, "Software process improvement through the Lean Measurement (SPI-LEAM) method," *Journal of Systems and Software*, vol. 83, no. 7, pp. 1275–1287, 2010.
- [2] J. Tian, "Measurement and continuous improvement of software reliability throughout software life-cycle," *Journal of Systems and Software*, vol. 47, no. 2-3, pp. 189–195, 1999.
- [3] A. Uskarcı and O. Demirörs, "Do staged maturity models result in organization-wide continuous process improvement? Insight from employees," *Computer Standards & Interfaces*, vol. 52, pp. 25–40, 2017.
- [4] A. A. Khan, M. Shameem, R. R. Kumar, S. Hussain, and X. Yan, "Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development," *Applied Soft Computing*, vol. 83, Article ID 105648, 2019.
- [5] C. Lassudrie and G. Gullà-Menez, "An experience in using risk management in a software process improvement programme," *Software Process: Improvement and Practice*, vol. 9, no. 1, pp. 3–12, 2004.
- [6] K. Kanoun, "A measurement-based framework for software reliability improvement," *Annals of Software Engineering*, vol. 11, no. 1, pp. 89–106, 2001.
- [7] S. Nazir, S. Shahzad, and S. B. S. Abid, "Selecting software design based on birthmark," *Life Science Journal*, vol. 11, no. 12, pp. 89–93, 2014.
- [8] S. Nazir, M. A. Khan, S. Anwar, H. Khan, and M. Nazir, "A novel fuzzy logic based software component selection modeling," in *Proceedings of the International Conference on Information Science and Application (ICISA)*, pp. 1–6, Suwon, Korea, May 2012.
- [9] M. Elliott, R. Dawson, and J. Edwards, "An evolutionary cultural-change approach to successful software process improvement," *Software Quality Journal*, vol. 17, no. 2, pp. 189–202, 2009.
- [10] T. Fukushima and S. Yamada, "Software process improvement by process monitoring and quality evaluation activities," *International Journal of System Assurance Engineering and Management*, vol. 1, no. 4, pp. 284–292, 2010.
- [11] T. R. D. Saputri and S.-W. Lee, "Integrated framework for incorporating sustainability design in software engineering life-cycle: an empirical study," *Information and Software Technology*, vol. 129, Article ID 106407, 2021.
- [12] M. Drury, K. Conboy, and K. Power, "Obstacles to decision making in Agile software development teams," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1239–1254, 2012.
- [13] F. J. Pino, F. García, and M. Piattini, "Software process improvement in small and medium software enterprises: a systematic review," *Software Quality Journal*, vol. 16, no. 2, pp. 237–261, 2008.
- [14] A. Börjesson, F. Martinsson, and M. Timmerås, "Agile improvement practices in software organizations," *European Journal of Information Systems*, vol. 15, no. 2, pp. 169–182, 2006.
- [15] J. Gasston and P. Halloran, "Continuous software process improvement requires organisational learning: an Australian case study," *Software Quality Journal*, vol. 8, no. 1, pp. 37–51, 1999.
- [16] K. K. Mohan, A. Srividya, H. Ravi, and R. K. Gedela, "Improvement of QoS in process centric software development and application of analytical network process (ANP) for the estimation of critical phases," *Opsearch*, vol. 45, no. 4, pp. 369–380, 2008.
- [17] Q. Zhang, Q. Ma, M. Liu, K. Zhong, B. Xu, and L. Wu, "Research on the software reliability quantitative evaluation of nuclear power plant digital control system based on non-homogeneous Poisson process model," *Annals of Nuclear Energy*, vol. 144, Article ID 107589, 2020.
- [18] C. C. Venters, R. Capilla, S. Betz et al., "Software sustainability: research and practice from a software architecture viewpoint," *Journal of Systems and Software*, vol. 138, pp. 174–188, 2018.
- [19] T. E. Colanzi, W. K. G. Assunção, S. R. Vergilio, P. R. Farah, and G. Guizzo, "The symposium on search-based software engineering: past, present and future," *Information and Software Technology*, vol. 127, Article ID 106372, 2020.
- [20] D. A. Chevers, A. M. Mills, E. W. Duggan, and S. E. Moore, "Toward a simplified software process improvement framework for small software development organizations," *Journal of Global Information Technology Management*, vol. 20, no. 2, pp. 110–130, 2017.
- [21] B. Cartaxo, G. Pinto, and S. Soares, "Towards a model to transfer knowledge from software engineering research to practice," *Information and Software Technology*, vol. 97, pp. 80–82, 2018.
- [22] J. J. M. Trienekens, R. J. Kusters, M. J. I. M. Van Genuchten, and H. Aerts, "Targets, drivers and metrics in software process improvement: results of a survey in a multinational organization," *Software Quality Journal*, vol. 15, no. 2, pp. 135–153, 2007.
- [23] T. Nonoyama, L. Wen, and T. Rout, "Current challenges and proposed software improvement process for VSEs in developing countries," in *Proceedings of the International Conference on Software Process Improvement and Capability Determination*, pp. 437–444, Dublin, Ireland, June 2016.
- [24] S. Nazir, S. Anwar, S. A. Khan et al., "Software component selection based on quality criteria using the analytic network process," *Abstract and Applied Analysis*, vol. 2014, Article ID 535970, 12 pages, 2014.
- [25] W. Shah, "Quality models," in *Software Product Quality Control* Springer, Berlin, Heidelberg, 2013.
- [26] S. Shahzad, S. Nazir, S. B. Abid, I. Zada, and R. Amin, "Software component project evaluation based on quality measure," *Life Science Journal*, vol. 11, no. 10, pp. 98–107, 2014, <http://www.lifesciencesite.com>.
- [27] International Standard, ISO/IEC 2382-36, "in Third edition," 2019-06.