

## **Elements of Quality Management in Software Process: A Formalised Approach**

**Kamal Uddin Sarker<sup>1</sup>, Azis Bin Deraman<sup>2</sup>**

<sup>1,2,3,4</sup> Faculty of Ocean Engineering Technology and Informatics, University Malaysia Terengganu, Malaysia

### **Abstract**

A software project is commenced to accomplish according to the plan and allocated budget that should satisfy users' requirements and integrate quality factors. But, till now significant numbers of software projects delivered without obligatory functionalities and qualities. Informal practice in communication, documentation, and management show weakness in time control, change control, and risk mitigation; as a result, the project exceeds time, budget, and scope. This paper proposes a formalized approach for software project management with embedded quality elements. This formalized approach is validated by 34 professionals and found that it is suitable for medium to large projects for any standard organization. In addition, an expert has compared the model by analytical hierarchy process with commonly used methodologies and found that it is better in terms of scope, schedule, risk, resources, documentation, instance change, quality, and human management besides sustainability practice. The quality management is formalized in such a way that encourages the practice of international standardization guidelines to continuous improvement of the project handling processes that would improve the quality of products.

**Keywords:** *software quality management, methodology, formalized approach, virtual management, sustainability practice*



This is an open access article under the CC-BY-NC license

### **INTRODUCTION**

Quality software is the outcome of long-term systematic work and implementation of a standard strategy (Xihui et al 2011) that is founded on the mission and vision of a software firm. So, a series of quality models, system development methodologies, and tools have been upgraded to enhance software quality and increase the success rate of soft product projects. The Standish Group has been publishing annual CHAOS (Create Havoc Around Our System) reports since 1995 and according to the CHAOS-2018, the success rate is 36%, challenges is 45%, and 19% of the projects failed (Johnson, J. 2020). All projects assume that will be successful project at the initiation stage but most of these become challenges during execution time and a few of those stopped without handover is called failed projects. A project becomes challenging due to the informal communication, documentation, and management practice so formal information specification and management are required for any software project (Sarker et al 2020). a challenged project stops in any phases of developing process without handover for the aforementioned reasons but not only for a single reason because there is causal relation among process, task, and people (Tim O. et al 2014). A software product could be handover without proper implementation of software quality factors: accuracy, effectiveness, portability, maintainability, sustainability, etc. due to the limitation in requirement analysis, defining scope, market analysis, and prediction of technological changes that demand a formalized approach (Kamal et al 2018). According to the ICT footprint (2018), the ICT sector will consume 15% of the total electricity consumption by 2040 due to increases in smart devices and

Corresponding author

Kamal Uddin Sarker, [ku\\_sarker@yahoo.com](mailto:ku_sarker@yahoo.com)

DOI: 10.31098/csetv1i2.473

Research Synergy Foundation

robotics. But, only 16% of developers consider power-saving features in their system (Pinto G et al, 2014). Green soft computing becomes one of the most important quality measures in sustainability because of increasing ICT subscribers (KU Sarker et al 2018). Quality factors identification and integration should be considered in requirement specification and implementation in design and development phases; finally quality will ensure by different levels of testing (Kamal Uddin Sarker et al 2020). On the other side, virtual groups are becoming popular due to the advent of low-cost internet and communication media thus software companies access skilled persons from any corner of the world (Ågerfalk et al, 2008). There are plenty of software development methodologies, standardization organizations, certification, and professional bodies are working along with academic researchers but till now

- A Remarkable Number of Software Projects Fail Every Year
- Limited Practice in Green Soft Computing
- No Special Methodology for Distributed Software Project Management

That demands a comprehensive model that ensures the quality of the product and process, guides to accomplish a project on time and budget, encourages virtual project management, and practice green computing.

## **LITERATURE REVIEW**

Standish Group (CHAOS-report, 2018) updated their modern project resolution benchmark for success factors: budget, plan, target, goal, valuable, and customer satisfaction; where the last four factors are related to the quality of the project, and the first two are mainly for business goals. It also classified failed projects (stopped before handover), challenging projects (a project that was delivered without ensuring quality or late, or over budget), and successful projects that meet user satisfaction and business goals. It identified 10 success factors of software projects: flexible methodology adequate by the organization, speed up the decision, team maturity, project sponsor, scope, optimization process, talent staff, standard architecture management environment, and user involvement. Varieties of system development methodologies (e.g. SP, Waterfall, Incremental, Spiral, RAD, V, Prototyping, Scrum, Agile, LS Scrum, etc.) are implanting based on the size, duration, risk, complexity, and type of a project; as well as different quality models (McCall, Boehm, ISO 9126-4/2001b, FURPS, IEEE, Dromey, etc.) are following to ensure quality. Bit till now the success rate is not satisfactory because of the new opportunities and challenges of technological advancement.

The latest global industry vision report of Huawei (2019) remarked that home robots will increase by 14%, AI and augment reality / virtual reality applications will be common for 5G network, 90% of smart device user will use intelligent personal assistance, “vehicle-to-everything technology” will be introduced by 15% vehicle, 103 can do work of 10k employees, 97% will use AI systems, enterprises will use 86% data, 85% business will move to cloud infrastructure, and software organizations need to manage huge amount of feature-driven information. It will create a vast demand for power and total consumption in the ICT sector was 3.9% to 4.6% from 2007 to 2012 (Salahuddin and Alam K, 2016), so green computing become an important topic and we included it in the software quality management methodology. Few works are progressing and a green software model is proposed by Kern (2013) that recommends selecting an energy efficiency platform, accurate and effective algorithms to reduce CPU cycles, proper memory allocation, and sustainable utilization of peripherals.

Distributed project management become popular due to the advancement of communication technology. Software industries have a boundary-less world to utilize skilled people for their projects that could reduce cost and time. A virtual team faces difficulties with different time zones, language, culture,

and documentation. Currently, there is no methodology to manage a virtual team for a distributed software project (Martinic et al, 2012) and in most cases, agile methodology is adopted (Sureshchandra and Shrinivasavadhani, 2008). A virtual model is proposed by Martinic (2012) by combining iterative and traditional approaches to reduce staffing cost the also recommends using effective tools for project monitoring and controlling. But there are no guidelines on how to overcome the challenges like language and documentation except suggestions for using video conferencing tools (Alawadi et al, 2015) to reduce technical debate. But we believe standard documentation could reduce ambiguity, technical debate, and language barrier.

This paper presents a quality model that formulized functionalities of system development methodologies in a structure that keeps control of the project and improves the quality of the product and process.

### **RESEARCH APPROACH**

The research proposed a quality model in section IV based on the problem statements of section II and requirements study of literature review. The model consists of the essential elements of quality management of software projects and needful actions to achieve business goals. Finally, it is validated by a set of professionals and to show the acceptance of quality elements that are integrated into the model. It is compared by an expert with a few popular existing methodologies in the software development life cycle by Analytical Hierarchy Process (AHP). And followed by result analysis and concluding remarks with future work.

### **Elements of Quality Management And a Formalized Approach**

Customers assign a software project to a soft computing firm that has specific goals and objectives that should achieve by this software. So, a firm can achieve its own business goals by fulfilling users' objectives.

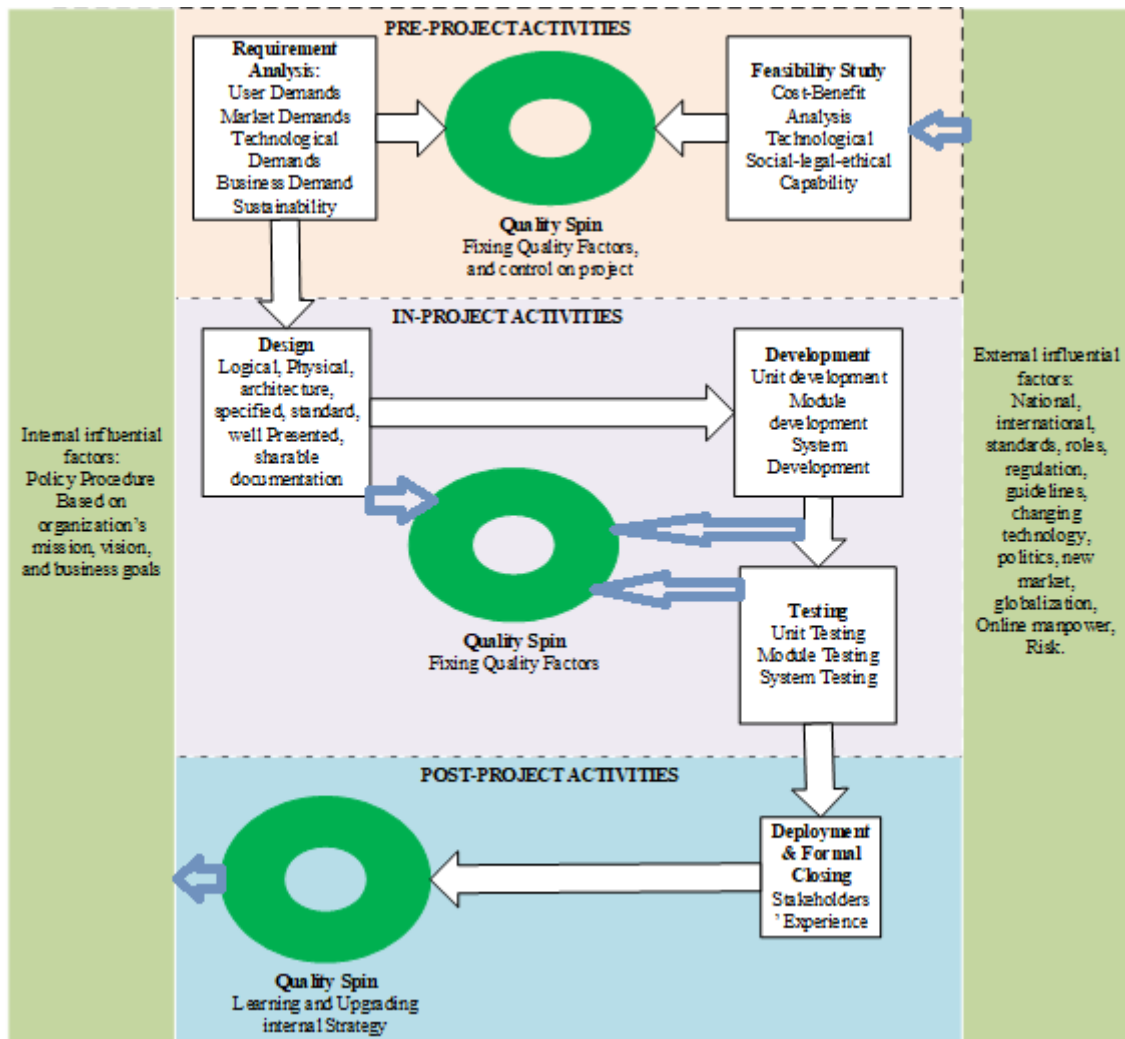


Figure 1. A Formal Software Quality Management Approach (FSQMA)

The architecture is divided into three layers: project initiation, project execution, and project closing are called post-project activities, in-project activities, and post-project activities respectively but all are an essential part of the proposed approach. Three quality spins are applied in three different layers because of the sequential flow of the architecture. Quality product and the process is not possible to get by taking some actions in a certain phase. It is related to internal and external influential factors (left and right side), aligned with the moto and business goal of an organization that could achieve by overcoming some internal and external challenges. The first quality spin (figure 2) is used to identify and integrate demanded quality factors based on the application; the second spin will integrate into the design and implement to develop, finally it will be confirmed by testing. In the closing phase, users' feedback and projects documentation will guide to update policy, roles-regulation, and the best practices.

The proposed approach is structured for continuous improvement in software project functionalities of a standard organization. It encourages following international standardization guidelines that could reduce the challenges of distributed management systems by practicing

controlled (descriptive logic, predicate logic, ontology, etc.) languages instead of generic language. Control language documentation will reduce ambiguity and technical debate; furthermore, a developer can easily convert a controlled language design to coding.

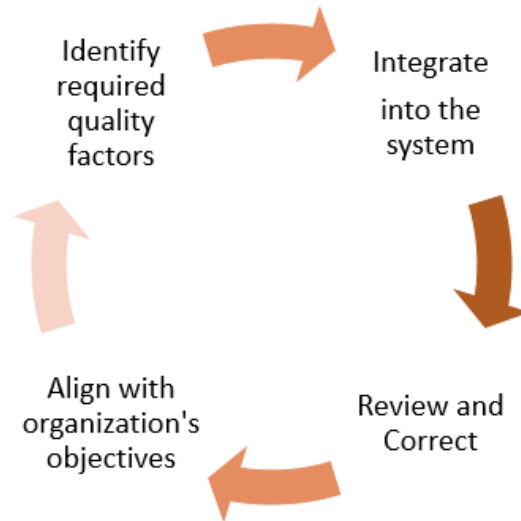


Figure 2. Spin structure of FSQMA

### Empirical Validation

The proposed FSQMA is evaluated by 34 professionals of Bangladesh in June-August 2020, who used the model for their software projects and provides online feedback (Google Docs). There were 9 distinguished measuring domains (scope, schedule, cost, quality, risk, management, documentation, resource, sustain) of the model and each domain carries 3 questions and an average score of three questions is the final score of the respective domain. A Likert scale (5= strongly agree, 4=agree, 3=neutral, disagree=2, and strongly disagree=1) is implemented to convert qualitative data to quantitative data for statistical analysis (heading of the appendix-A). 34 Participants attend in the evaluation process who have 1-15 years working experience in software industries where 35% have experience in methodology adaptation of software projects while 65% apply a methodology that is selected by the organization. Professionals were attended from 28 organizations where 74% have more than 5 years of professional experience with multiple areas (system analysis, design, development, testing) of software projects. 57.6% prefer ISO standardization guidelines and 9.1% follow both ISO and CMMI level-3 recommendations for their project but 27.3% do not focus on any standardization process. The FSQMA is good for standard organizations that can implement medium to large projects and they need initial labor to implement only; the overview of the reflection is given in Table-1.

Table 1.

Acceptance of the Proposed FSQMA

Items	Criteria	Feedback
The FSQMA is appropriate for (Based on Project Size)	Large Size Project	6%
	Medium to Large Size Project	47%
	Small to Medium Size Project	22%
	All Size Project	25%
The FSQMA is appropriate for (Based on Organization)	Standard Organization	44%
	Any software Firms	53%
	Independent Developer	3%
The complexity of the FSQMA	Easy to understand	61%
	As usual	3%
	Difficult to understand	36%

Table 2. shows the statistical analysis of the professionals' feedback for the proposed FSQMA. It gives equal importance among 9 items (see table II), out of 1 the relative importance is  $>.8$  and  $<.87$  where respondents feel that it is more planned management (schedule has higher value  $0.867+$ ) and comparatively less importance shows for resource management ( $0.81+$ ). The mean value for all criteria are in between agree and strongly agree ( $>4$ ). From the aforementioned study, it is acceptable for the project where critical management is required in the micro-level of management. It promotes sustainability in a software project and into the product, helps to ensure quality, tracks on schedule and budget, supports standard documentation for management process, task, resources, scope, and risk.

Table 2.

Relatively Importance of the Formalized Approach

<b>Factor</b>	<b>Mean</b>	<b>Standard Deviation</b>	<b>Relatively Importance</b>
<i>Scope</i>	4.17627451	3.277605444	0.835254902
<i>Schedule</i>	4.338308458	2.965447253	0.867661692
<i>Cost</i>	4.194666667	3.042841181	0.838933333
<i>Quality</i>	4.171457086	3.205718137	0.834291417
<i>Management</i>	4.28749259	3.061097203	0.857498518
<i>Documentation</i>	4.140250627	3.816912807	0.828050125
<i>Risk</i>	4.117647059	3.19976295	0.823529412
<i>Resource</i>	4.059259259	3.345624648	0.811851852
<i>Sustainability</i>	4.128395062	3.360460802	0.825679012

### Expert Confirmation

The FSQMA is compared with commonly used methodologies for 11 features and Analytical Hierarchy Process (AHP) is applied for pair comparison. Mr. Abdullah Al Muqim (lead developer and project manager, Coca-Cola, Atlanta, Georgia, USA) who is experienced in software project development and management performs pair comparison by AHP. The comparison project management factors: documentation, process, task, resource, risk, human, functional requirement, non-functional requirement, distributed, sustainability, instant are split into fixed numbers of sub-criteria for better comparative study. A comparison table is used to accept feedback like table 3; for example, model-1 is very strongly preferable to model-2, model-3 is extremely preferable to model-1, and model-2 is similarly important than model-3. The feedback that is collected by table 3. is converted to an AHP scale (9 for extremely preferable = XP, 7 for very strongly preferable = VSP, 5 for strongly preferable = SP, 3 for moderate preferable = MP, and 1 for similar importance = SI) for analysis. If model-1 is VSP then model-2, hence the value for model-1 is 7 and model-2 is 1/7; similarly, model-1 has 1/9 for the 9 of model-3; and model-2 and model-3 have 1. A reciprocal matrix is developed from the expert's data (left side matrix of table 4.) that is normalized by dividing each cell value by the total of each column (Middle matrix of table 4); finally, an AHP rank is generated (average of each row) called criteria weight (rightmost column in table 4.) and proposed FSQMA got the highest ranking for all measuring factors. For example, in table 4 expert's opinion is formalized FSQMA is strongly preferable to agile and PRINCE2 and very strongly preferable to waterfall; PRINCE2 is equally important to agile and waterfall; and agile is strongly preferable to the waterfall that is mapped to a reciprocal matrix.

Table 3.

AHP Data Collection Format

	XP	VSP	SP	MP	SI	MP	SP	VSP	XP	
Model-1		√								Model-2
Model-1									√	Model-3
Model-2					√					Model-3

Table 4.

Example AHP Calculation

Reciprocal Matrix for Criteria-1 of Documentation					Normalized Matrix for Criteria-1 of Documentation					Comparative Weighted	
	FSQMA	PRINCE2	Waterfall	Agile		FSQMA	PRINCE2	Waterfall	Agile		
FSQMA	1.00	5.00	7.00	5.00	FSQMA	0.65	0.63	0.50	0.69	FSQMA	0.62
PRINCE2	0.20	1.00	1.00	1.00	PRINCE2	0.13	0.13	0.70	0.14	PRINCE2	0.12
Waterfall	0.14	1.00	1.00	0.20	Waterfall	0.09	0.13	0.07	0.03	Waterfall	0.08
Agile	0.20	1.00	1.00	0.50	Agile	0.13	0.13	0.36	0.14	Agile	0.19

## CONCLUSION

Quality factors are requirements that should incorporate in the right phase and confirm in the following phases. So proposed approach carries three distinguished quality-spin to review, update, confirm and recommend if need. The paper proposed a generalized and flexible structure but has finite states with start and end. It can moderate the internal blocks and an organization can add activity in between two stages for any special project. Role-based distributed project management tools can be adequate based on the application that will help to manage virtual project teams. It is not only focused on the testing process (unit, module, and integrated) to ensure quality but also links quality factors with all phases and sustainability brings to strategy level.

For explicit specification, it will minimize scope creeping; standard documentation will reduce technical debates and improve the instant decision; micromanagement and tracking facility keeps control on cost and time; risk, process, and task management would be efficient due to the formal management approach; standard documentation will improve reusability and sustainability.

Most of the computer / mobile applications will consist of AI, data analysis, IoT, and machine learning processes soon that will demand information specification and the proposed model will be the right choice for the software farms. It is not recommended for organizations with short-term goals or individual application developers due to the explicit specification nature of the model. In the future, we will develop logic, rules, and axioms for project information specification from where automatic ontology could be extracted. Second, the information cannot be acquired in a timely manner. Because the majority of respondents are unwilling or unable to finish the survey, this is the case. Researchers will find it more difficult to acquire data as a result of this.



## ACKNOWLEDGEMENT

The author would like to acknowledge the Malaysia Ministry of Education and Research Management Center at Universiti Malaysia Terengganu (UMT) for funding the grant with the reference code FRGS 59561 to pursue research in this field.

## REFERENCES

- Alawadi, Mohd & Hammad, Mustafa & Alhasanat, Esra'a. (2015). Dynamic Virtual Teams in Project Management. *International Journal of u- and e-Service, Science and Technology*. 8. 10.14257/ijunesst.2015.8.7.25.
- CHAOS-report (2018). Standish Group 2018 Chaos Report - Q&A With Jennifer Lynch, visited: 9th July 2021, available <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
- Global Industry vision (2019). Touching An Intelligent Future: Explore the megatrends shaping the future toward 2025.visited: 20 August 2019, available:<https://www.huawei.com/en/pressevents/news/2019/8/huawei-predicts-10-megatrends-2025>
- ICT footprint (2018). Decreasing ICT energy consumption – the power of data centres and people’s will. visited: 10 July 2019, available: <https://ictfootprint.eu/en/webinar/decreasing-ictenergy-consumption-%E2%80%93-power-data-centres-and-people%E2%80%99s-will>.
- Johnson, J. (2020). Review CHAOS Report 2018. [online] On Portfolio, Programme and Project Management. Available at: <https://hennyporntman.wordpress.com/2020/01/03/review-chaos-report-2018/> [Accessed 20 Oct. 2021].
- K U Sarker, Dr. Aziz Deraman, Raza Hasan, (2018) “Green Soft Computing”, Conference: ICEL3S2018, Feb 2018, Malacca, Malaysia;, Semiannual ISSN: 1112-9867, issue-65 *Journal Of Fundamental And Applied Sciences*, volume-10, page-462-470 <http://jfas.info/psjfas/index.php/jfas/article/view/4280>
- K. Sureshchandra and J. Shrinivasavadhani, (2008) “Adopting Agile in Distributed Development,” in *Proc. 3rd International Conf. on Global Software Engineering*, Bangalore, India, 2008, pp. 217-221.
- Kamal Uddin Sarker, Aziz Bin Deraman, Raza Hasan and Ali Abbas, (2020). “SQ-Framework for Improving Sustainability and Quality into Software Product and Process” *International Journal of Advanced Computer Science and Applications(IJACSA)*, 11(9). <http://dx.doi.org/10.14569/IJACSA.2020.0110909>
- Kamal Uddin Sarker, Dr. Aziz Deraman, Raza Hasan, Abdul Hadi Bhatti, (2018). “Software Development Life Cycle Developing Framework” 5th International Conference on Communication and Computer Engineering (ICOCOE) Melaka, Malaysia. Conference Proceeding Page 267-271 [https://maltesas.my/msys/explore/pubs/1562833286/1562833286\\_Article\\_1564562131.pdf](https://maltesas.my/msys/explore/pubs/1562833286/1562833286_Article_1564562131.pdf)
- Kern, Eva & Dick, Markus & Naumann, Stefan & Guldner, Achim & Johann, Timo. (2013). Green software and green software engineering - definitions, measurements, and quality aspects. the First International Conference on Information and Communication Technologies for Sustainability, held at ETH ZurichConference Proceedings, page: 87-94
- Martinic, K. Fertalj, and D. Kalpic (2012). Methodological Framework for Virtual Team Project Management. *International Journal of Innovation, Management and Technology*, Vol. 3, No. 6.
- P. J. Ågerfalk, B. Fitzgerald, H. Holmström Olsson, and E. Ó Conchúir, (2008) "Benefits of Global Software Development: The Known and Unknown," presented at International Conference on Software Process, ICSP 2008, Leipzig, Germany.

Pinto G, Castor F, Liu Y D (2014). Mining questions about software energy consumption. In ACM 11th Working Conference on Mining Software Repositories, 2014, pp. 22-31

Salahuddin and Alam K. (2016). Information and Communication Technology, Electricity Consumption and Economic Growth in OECD Countries: A Panel Data Analysis. *Int. J. Electr. Power Energy Syst.* 2016, 76, 185–193.

Sarker KU, Deraman A, Hasan R, Abbas A. (2020) Explicit specification framework to manage software project effectively. *Indian Journal of Science and Technology.* 13(36): 3785-3800. <https://doi.org/10.17485/IJST/v13i36.1244>

Tim O. Lehtinen, M., ntyl, M. V., Vanhanen J., Lassenius C. and Itkonen J. (2014), Perceived Causes of Software Failure-An analysis of their relationships, *Journal of Information and Software Technology* 56, page-623-643“

Xihui Zhang, Sandra, Michael, Mark L. Gillenson (2011). Engaging Testers Early And The Oughout The Software Development Process: Six Models And A Simulation Study. *Journal of Information Technology Management*, Volume XXII, Number 1, 2011