UMT PRESS

Journal of Mathematical Sciences and Informatics
Journal Homepage: https://journal.umt.edu.my/index.php/jmsi
eISSN: 2948-3697
DOI: http://doi.org/10.46754/jmsi.2025.06.006

JMSI
Journal of Mathematical Sciences and Informatics

SOFTWARE MAINTENANCE ASSESSMENT: AN ANALYSIS OF DETERMINATION FACTORS

KU SAIMAH KU IBRAHIM[1]*, ZULKEFLI MANSOR[2], JAMAIAH YAHAYA[3] AND AZIZ DERAMAN[4]

[1]Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, Malaysia. [2]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 46300 Bangi, Selangor, Malaysia. [3]The National Council of Professors Global Reach, Country Heights, 43000 Kajang, Selangor, Malaysia. [4]Faculty of Computer Science and Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia.

**Corresponding author: saimah@umpsa.edu.my*

ARTICLE INFO

Article History:
Received 24 May 2024
Revised 14 April 2025
Accepted 14 April 2025
Published 15 June 2025

Keywords:
Software maintenance;
software maintenance assessment;
software process assessment;
assessment factor;
determination factor.

ABSTRACT

Software maintenance is the longest and most expensive Software Development Life Cycle (SDLC) phase. It contributes a major part of a software system's total life cycle cost. An effective software maintenance assessment is crucial to ensure the software system's longevity and efficiency. The effective assessment will provide valuable insight into the decision-making guidance and foster continuous improvement. However, the factors that influence the assessment process are not ideally mapped with the existing software maintenance process. To provide effective planning and resource allocation, the software maintenance team needs to determine the factors that influence the process. This paper presents a review of existing literature to identify and categorise the determination factors that influence the software maintenance assessment. Literature studies show that the key factors can be identified and grouped into several categories. The analysis reveals factors that influence the software maintenance assessment, which is categorised into project characteristics, development teams' skills, client involvement, project constraint, organisational culture, and documentation. The determination factors will help the software maintenance team assess the priorities of their efforts and allocate the available resources efficiently. This paper contributes valuable knowledge on the assessment process and offers useful guidance for practitioners looking to improve their software maintenance efforts.

2020 Mathematics Subject Classification:

Introduction

Software maintenance is a continuous process to ensure the longevity, reliability, and efficiency of a software application. It is also considered the longest and most expensive phase in the software lifecycle [1]. In many literatures, software maintenance concerns the modification made to the software application once it is delivered to the end users. The software maintenance category can be classified into corrective maintenance, perfective maintenance, preventive maintenance, and adaptive maintenance [2, 3, 4, 5]. Those classifications are based on the activities done to keep the software operational and satisfy the user's requirements.

The software maintenance process comprises several activities. The activities range from bug fixing and performance enhancements to adaptations of software applications to changing requirements or environments [6, 7]. The classification of software maintenance according to its category is based on the activities done during the management of the legacy system. However, the existing factors are not adequate in guiding practitioners during the decision-making process [8, 1]. Current technological advancements should be observed during software maintenance activities to propose the most suitable model for both practitioners and researchers. Those activities are important to ensure the software system keeps functioning according to the users requirements.

Due to the constant change in software applications during the software maintenance phase, the program structures tend to degrade. Over time, users keep on request to add more functionality to the current system. On the other hand, stakeholders will see that the expansion of the system will provide more benefits to the users. The process of implementing change might introduce new bugs to the software and cause the internal structure to deteriorate [9, 10, 11]. To overcome this issue, maintainers need to understand the activities involved during the software maintenance process clearly. The software maintenance process is shown in Figure 1.

Software maintenance is different from software development as it starts at the clients request. The work starts upon the team receiving the client's request [12, 13]. Usually, the client request is in a form called a Change Request (CR) or Modification Request (MR). The form consists of the necessary information about the change or a new request by the users or stakeholders.

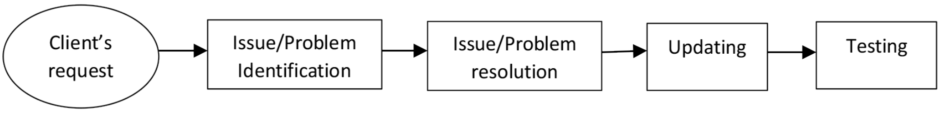

Figure 1: Software maintenance process life cycle [6]

Before facilitating the process of assessing the type of software maintenance, it is crucial to address the factors that influence the determination process. The factors need to be defined earlier so the next process would become easier for the team. Based on the identified factors, the software maintenance engineer will be able to plan and allocate resources in advance. Thus, this study aims to elicit and propose the factors that contribute to the software maintenance assessment. By providing the factors, it will help industry practitioners to consider the identified factors at an early stage.

Methodology

This section explains the process used to identify the factors that contribute to the determination of software maintenance assessment. There are phases involved, and the first phase is to investigate those factors in the existing literature. Figure 3 presents the proposed process involved in the research.

Phase 1: Literature Study

This study performed document analysis to accomplish the aforementioned objectives. Document analysis is a systematic procedure for reviewing or evaluating existing material, either printed or electronic [14]. The process of document analysis involved skimming, reading, and interpretation. Figure 2 illustrates the steps involved in the first phase of this study.

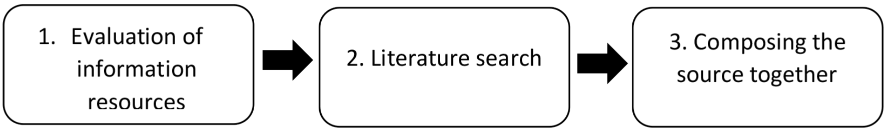
1. Evaluation of information resources
2. Literature search
3. Composing the source together

Figure 2: Research design

The initial step is to evaluate the information resources that come from various sources, including online databases and books, either printed or electronic. The credibility of the sources is thoroughly checked to ensure that those documents are retrieved from a valid database.

Next, during the literature search in step two, different keywords were applied, and different sources were gained. In this study, several databases were referred to retrieve the information: IEEE Xplore, Scopus, Science Direct, and Google Scholar. Table 1 presents a summary of the databases used during the literature search activity.

Table 1: Online database for literature search

Indexing Database	Number of Studies Retrieved
IEEE Xplore	6
Scopus	7
Science Direct	6
Google Scholar	4
ACM Digital Library	3

The last step involved composing the sources together, where all the selected sources for document analysis are analysed. This activity helps to figure out the factors that contribute to the assessment of software maintenance classification. Each finding is then divided into several categories, which later represent the component in the software maintenance assessment model.

This extensive literature review, employed during document analysis, helps to identify the factors studied by various researchers. Then, the list of factors related to the software maintenance assessment is identified and sorted out. Next, the redundant factors are removed.

Phase 2: Validation of Proposed Factor

Phase two of this study will be related to the validation of the proposed factors through expert validation. The expert panel involved a group of experts in the fields of software maintenance and software development. They should have experience in maintaining a legacy system in their organisations for some time. Based on [15, 16], it is possible to have at least three expert panels to evaluate the proposed factors. For this phase, those three expert panels should be enough to gain a brief idea and support the software maintenance assessment model.

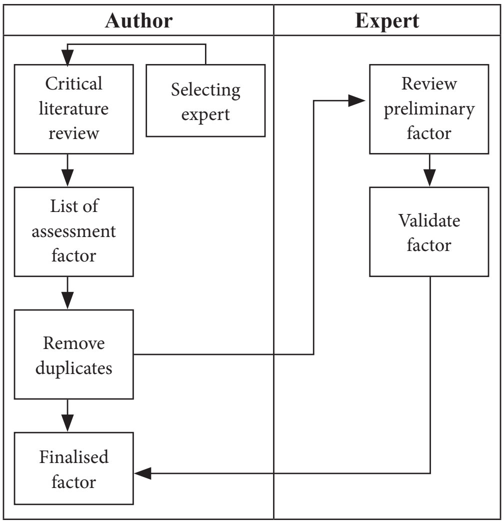
Author
Expert
Critical literature review
Selecting expert
List of assessment factor
Remove duplicates
Finalised factor
Review preliminary factor
Validate factor

Figure 3: Proposed process in factors validation

The proposed approach aims to develop reliable factors that help to determine the software maintenance assessment. The expert panel's point of view is crucial to validating the findings based on industrial perspectives. Therefore, these factors are crucial to this study in supporting the development of a software maintenance assessment model.

Results and Discussions

In this study, the results from the document analysis process have helped to identify the factors that influence the software maintenance assessment. All factors identified during the document analysis process are listed, and thematic analysis is performed to classify those factors into relevant categories.

Table 2 presents the identified factors that influence software maintenance assessment in general. According to the literature, these factors may vary depending on the type of application [17], [1], [18]. In this study, the identified factors are listed to give a brief idea of how they will affect the assessment of software maintenance. These factors will be used as a guideline for software practitioners to forecast the preparation needed for the assessed software maintenance project.

Table 2: Software maintenance assessment: determination factors

Factor	Source
Size	[19], [6], [9], [2], [20], [21], [22]
Criticality	[4], [6], [9], [22]
Communication skills	[23], [5], [8], [12], [20],
Team competency	[19], [1], [23], [24], [4], [5], [8], [21]
Knowledge domain	[1], [23], [24], [12], [21]
Collaboration	[23], [4], [6], [8], [12], [20]
Commitment	[23], [24], [6], [8], [20]

Cost	[23], [6], [7], [8], [25]
Time	[24], [4], [7], [2], [26]
Quality	[6], [2], [25], [26]
Hierarchy	[23], [2], [8], [9]
Obsolete document	[19], [27]
System specification	[19], [28], [27]
Documentation	[19], [29], [24], [9], [25], [20], [28], [27]

Based on the document review conducted, 14 factors have been figured out. Documentation and team competency factors were found in eight kinds of literature. It shows that these two factors are important to consider when assessing the software maintenance project. Seven kinds of literature emphasise the size of the project as one of the factors that need to be considered during the assessment process. Normally, the size of a software project is measured by the Line of Code (LoC), number of sub-systems, and number of users [2].

Another factor was collaboration between the customer and the maintenance team. Researchers have discussed this factor in six studies conducted in previous literature. Four articles during the document analysis discussed communication skills, knowledge domain, commitment, cost, and time factors. Four articles found more factors, such as criticality, quality, and hierarchy, while system specification and obsolete documents were discussed in three and two articles, respectively.

The identified factors are then classified into one significant main factor that will later represent a component in the future proposed model. We employed coding techniques for the defined factors and grouped the factors with similar intentions under relevant categories. Each component will then create a linkage between another component to complete the model. Table 3 presents the results after all the factors have been grouped into their main factor.

Table 3: Factor and sub-factor classification

Factor	Sub-Factors
Project Characteristics	Size, criticality
Development teams' skills	Communication skills, team competency, knowledge domain
Client involvement	Collaboration, commitment
Project constraint	Cost, time, quality
Organisational culture	Collaboration, hierarchy
Documentation	Updated, obsolete

The objective of this study has been achieve through the successfully conducted document analysis process. Six factors have been identified together with its sub-factors. Project characteristics have been seen as important factors to be considered during the assessment process [25], [20]. The sub-factor is followed by the size of the software application that needs to be considered and the criticality of the project. Other factors are the development team's skills, client involvement, project constraints, organisational culture, and documentation, which are crucial to the software application.

Future work is then scheduled to validate the factors and sub-factors by the expert in the industry. The expert panels are among the project managers with more than 14 years of experience in managing the legacy system[30]. Based on the years of experience, the expert panel might have been involved in many projects related to legacy system modernisation. Their insights are valuable in helping the finalisation of the factors that have been extracted in the literature review.

Conclusions

This study was conducted to identify the factors that affect the assessment of the software maintenance process. The existing factors in the previous literature do not ideally map with the current process in software maintenance, especially in managing the legacy system application. In addition, based on the latest study conducted, we have identified fourteen (14) factors, including size, criticality, communication skills, team competency, knowledge domain, collaboration, commitment, cost, time, quality, hierarchy, obsolete document, system specification, and documentation. Those factors have been grouped into categories such as project characteristics, development team skills, client involvement, project constraint, organisational culture, and documentation.

The study found that the defined factors significantly impact software maintenance assessment. Practitioners in the software industry will use the identified factors as their guidelines to assess the process of software maintenance assessment. The results will further contribute to the development of a software maintenance assessment model.

In future work, the expert panel will validate the factors and sub-factors through interview sessions to finalise the evaluation and refinement of the defined assessment factor. The output will then be used to establish the component for the software maintenance assessment model.

Acknowledgements

The author would like to thank the National University of Malaysia (UKM) and Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA) for supporting this research.

Conflicts of Interest Statement

The authors declare that they have no conflict of interest.

References

[1] Almogahed, A., Mahdin, H., Omar, M., Zakaria, N. H., Mostafa, S. A., AlQahtani, S. A., Pathak, P., Shaharudin, S. M., & Hidayat, R. (2023). A refactoring classification framework for efficient software maintenance. *IEEE Access*, *11*, 78904-78917. https://doi.org/10.1109/access.2023.3298678

[2] Ikram, A., Riaz, H., & Salman Khan, A. (2018). Eliciting theory of software maintenance outsourcing process: A systematic literature review. *International Journal of Computer Science and Network Security*, *18*(4), 132-143.

[3] Rehman, F. U., Maqbool, B., Riaz, M. Q., Qamar, U., & Abbas, M. (2018). Scrum software maintenance model: Efficient software maintenance in agile methodology. *2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 2018* (pp. 1-5). https://doi.org/10.1109/ncg.2018.8593152

[4] Levin, S., & Yehudai, A. (2019). Visually exploring software maintenance activities. *2019 Working Conference on Software Visualization (VISSOFT), Cleveland, OH, USA, 2019* (pp. 110-114). https://doi.org/10.1109/VISSOFT.2019.00021

[5] Ikram, A., Jail, M. A., Ngah, A., & Khan, A. S. (2020). Towards offshore software maintenance outsourcing process model. *International Journal of Computer Science and Network Security*, *20*(4), 6-14.

[6] Almashhadani, M., Mishra, A., Yazici, A., & Younas, M. (2023). Challenges in agile software maintenance for local and global development: An empirical assessment. *Information*, *14*(5), Article 261. https://doi.org/10.3390/info14050261

[7] Araujo, J., Melo, C., Oliveira, F., Pereira, P. A. A., & Matos, R. (2021). A software maintenance methodology: An approach applied to software aging. *2022 IEEE International Systems Conference (SysCon)*, 1-8. https://doi.org/10.1109/syscon48628.2021.9447082

[8] Stojanov, Z. (2021). Software maintenance improvement in small software companies: Reflections on experiences. *CEUR Workshop Proceedings*, *vol. 2913*, pp. 182-197. https://doi.org/10.47350/iccs-de.2021.14

[9] Fernández-Sáez, A. M., Chaudron, M. R. V., & Genero, M. (2018). An industrial case study on the use of UML in software maintenance and its perceived benefits and hurdles. *Empirical Software Engineering*, *23*(6), 3281-3345. https://doi.org/10.1007/s10664-018-9599-4

[10] Hassan, S. I., & Khan, A. S. (2017). Eliciting theory for software maintenance SLA management framework. *2017 International Conference on Frontiers of Information Technology (FIT)*, *Islamabad*, *Pakistan* (pp. 241-246). https://doi.org/10.1109/fit.2017.00050

[11] Ali, S. S., Zafar, M. S., & Saeed, M. T. (2020). Effort estimation problems in software maintenance - A survey. *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, *January 2020*. https://doi.org/10.1109/icomet48670.2020.9073823

[12] Masrat, A., Makki, M. A., & Gawde, H. (2021). Software maintenance models and processes: An overview. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3838444

[13] Stojanov, Z., Dobrilovic, D., & Stojanov, J. (2013). Analyzing trends for maintenance request process assessment: Empirical investigation in a very small software company. *Theory and Applications of Mathematics & Computer Science*, 3(2), 59-74.

[14] Morgan, H. (2022). Conducting a qualitative document analysis. *The Qualitative Report*, *27*(1), 64-77. https://doi.org/10.46743/2160-3715/2022.5044

[15] Baharuddin, M. F., Masrek, M. N., Shuhidan, S. M., Razali, M. H. B. H., & Rahman, M. S. (2021). Evaluating the content validity of digital literacy instrument for school teachers in Malaysia through expert judgement. *International Journal of Emerging Technology and Advanced Engineering*, *11*(7), 71-78. https://doi.org/10.46338/ijetae0721_09

[16] Beecham, S., Hall, T., Britton, C., Cottee, M., & Rainer, A. (2004). Using an expert panel to validate a requirements process improvement model. *Journal of Systems and Software*, *76*(3), 251-275. https://doi.org/10.1016/j.jss.2004.06.004

[17] Ramanujan, S., & Nerur, S. (2009). An exploratory analysis of the state of software maintenance research: An author co-citation analysis. *Journal of Systems and Information Technology*, *11*(2), 117-130. https://doi.org/10.1108/13287260910955093

[18] Hönel, S., Ericsson, M., Löwe, W., & Wingkvist, A. (2020). Using source code density to improve the accuracy of automatic commit classification into maintenance activities. *Journal of Systems and Software*, *168*, Article 110673. https://doi.org/10.1016/j.jss.2020.110673

[19] Ulziit, B., Warraich, Z. A., Gencel, C., & Petersen, K. (2015). A conceptual framework of challenges and solutions for managing global software maintenance. *Journal of Software Evolution and Process*, *27*(10), 763-792. https://doi.org/10.1002/smr.1720

[20] Nguyen, V., Boehm, B., & Danphitsanuphan, P. (2010). A controlled experiment in assessing and estimating software maintenance tasks. *Information and Software Technology*, *53*(6), 682-691. https://doi.org/10.1016/j.infsof.2010.11.003

[21] Ferreira, M., Bigonha, M., & Ferreira, K. A. M. (2021). On the gap between software maintenance theory and practitioners' approaches. *Proceedings 2021 IEEE/ACM 8th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)*, *Madrid*, *Spain* (pp. 41-48). https://doi.org/10.1109/ser-ip52554.2021.00015

[22] Rahman, H. U., Raza, M., Alzayed, A., Afsar, P., Alharbi, A., Alosaimi, W., & Khan, U. H. (2022). OffshoringDSS: An automated tool of application maintenance offshoring. *Applied Sciences*, *12*(21), Article 10913. https://doi.org/10.3390/app122110913

[23] Al Rababah, A., & Alzahrani, A. A. (2019). Software Maintenance Model through the Development Distinct Stages. *International Journal of Computer Science and Network Security*, *19*(2), 23-28. http://paper.ijcsns.org/07_book/201902/20190204.pdf

[24] Demelo, A., & Sanchez, A. (2006). Software maintenance project delays prediction using Bayesian Networks. *Expert Systems With Applications*, *34*(2), 908-919. https://doi.org/10.1016/j.eswa.2006.10.040

[25] Wang, A. Y., Wang, D., Drozdal, J., Muller, M., Park, S., Weisz, J. D., Liu, X., Wu, L., & Dugan, C. (2022). Documentation matters: Human-Centered AI system to assist data science code documentation in computational notebooks. *ACM Transactions on Computer-Human Interaction*, *29*(2), 1-33. https://doi.org/10.1145/3489465

[26] Khan, J. Y., & Uddin, G. (2022). Automatic Code Documentation Generation Using GPT-3. *ACM International Conference on Automated Software Engineering*. https://doi.org/10.1145/3551349.3559548

[27] Aghajani, E., Nagy, C., Linares-Vásquez, M., Moreno, L., Bavota, G., Lanza, M., & Shepherd, D. C. (2020). Software documentation: The practitioners' perspective. ICSE '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (pp. 590-601). https://doi.org/10.1145/3377811.3380405

[28] Bourrie, D. M., Cegielski, C. G., Jones-Farmer, L. A., & Sankar, C. S. (2014). Identifying characteristics of dissemination success using an expert panel. *Decision Sciences Journal of Innovative Education*, *12*(4), 357-380. https://doi.org/10.1111/dsji.12049