Investigating the Impact of Software Maintenance Activities on Software Quality: Case Study

VP Pamunuwa[1#], DP Deraniyagala[1#], VTB Kulasekara[1#], RDAV Thennakoon[1#] and BNS Lankasena[2]

[1#]*Department of Computer Science, Faculty of Computing, General Sir John Kotelawala Defence University, Rathmalana, Sri Lanka*

[2]*Department of Information & Communication Technology, Faculty of Technology, University of Sri Jayewardenepura*

[1#]37-se-0002@kdu.ac.lk,

Abstract - *Software maintenance is crucial for the reliability, functionality, and satisfaction of software systems. Although it might be expensive to keep software in good condition, it is essential to keep the software maintenance expenditure to a minimum without sacrificing software quality. Based on two leading software development organizations in Sri Lanka, the study examines how software maintenance operations affect software quality and identifies ways to reduce maintenance expenses without compromising quality. A comprehensive literature review was undertaken to discern a compelling research problem that would serve as the focal point for the study. The study was conducted using structured interviews with senior and operational staff from two organizations to quantify the impact of maintenance procedures on software quality preservation and proactively identify effective strategies. Both organizations adopt a proactive approach to software maintenance, encompassing bug fixing, updates, enhancements, and security updates while employing testing, quality assurance, monitoring, user feedback, and defect tracking to measure the impact of maintenance activities. Additionally, they predominantly utilize automated deployment, continuous integration, continuous deployment (CI/CD), and cloud-based deployment in their software deployment practices, with some adoption of containerization (e.g., Docker) as well. The findings show that software maintenance is essential, and many tasks are carried out to maintain quality, including testing, monitoring, user input, and defect tracking. Future studies should concentrate on creating more efficient maintenance methods to save expenses while maintaining high-quality software. This evaluation offers knowledge that practitioners may use to create efficient maintenance plans for software systems.*

Keywords— software maintenance, software quality, cost reduction

I. INTRODUCTION

Software maintenance is a crucial part of developing software that may have a big influence on the final product's quality. The examination of the literature demonstrates that while corrective, adaptive, and perfective maintenance positively enhance software quality, delayed maintenance or no maintenance at all might have a negative impact on it. Particularly perfective and corrective maintenance practices have a significant impact on the caliber of software. The actual data demonstrates that perfective maintenance, followed by corrective maintenance, has the greatest beneficial effects on software quality.

The study also emphasizes how improperly performed maintenance chores have the potential to result in flaws and other issues, thus decreasing the product's quality. Additionally, due to changing technology and customer needs, software maintenance is tough. To overcome these obstacles and ensure software quality, proper planning and administration are essential. Giving software maintenance operations significant attention and financial support is crucial to ensuring that program quality is maintained over time.

II. LITERATURE REVIEW

Software engineering research has been interested in the effect of maintenance efforts on software quality. Grubb and Takang's study, which highlights numerous maintenance tasks and their influence on software quality, is one of several studies that have looked at this issue. They come to the conclusion that while postponed maintenance or no maintenance at all might have a detrimental influence on software quality [1], corrective, adaptive, and perfective maintenance favorably impact software quality.

Corrective maintenance has a considerable influence on software stability, maintainability, and usability, according to Riaz et al.'s investigation into the topic [2]. Similar to this, Zhang and Li used data from a large-scale software project to examine the impact of maintenance operations on software quality and discovered that although adaptive maintenance has a negative impact, corrective and perfective maintenance has favorable effects [3].

In a case study on a university software system, Martnez Fernández et al. discovered that although adaptive maintenance had a detrimental effect on software quality, corrective and perfective maintenance have favorable effects [4]. In their last empirical investigation on a software system, Ushakova and Kashevnik discovered that while adaptive maintenance has a detrimental effect on software quality, corrective and perfective maintenance have favorable effects [5].

According to the examined research, there is broad agreement that software maintenance efforts improve the quality of the program. According to the research, perfective and corrective maintenance procedures in particular have a big influence on software quality.

In their empirical analysis of a software system, Kumar et al. (2019) discovered that perfective maintenance had the highest favorable influence on software quality [6]. Similarly, to this, Saliu and Oladele (2020) carried out a thorough assessment and discovered that perfective maintenance had the most favorable effects on software quality [7].

After reviewing empirical data, Rathore and Jaiswal (2020) discovered that corrective maintenance had the most favorable effects on software quality [8]. In their empirical investigation of a software system, Bhagat et al. (2019) provided more evidence for this conclusion [9]. Through a case study, Mubeen et al. (2019) looked at how software maintenance activities affected software quality and discovered that perfective maintenance had the most impact [10].

A software product that is delivered on time, within budget, and executes its functions correctly and efficiently may still have issues like being overly machine-dependent or difficult to integrate with other programs. It may also be hard to understand, hard to modify, difficult to use, or easy to misuse. Making appropriate design tradeoffs between development costs and operational costs, choosing software packages that can be easily adapted to changing needs and hardware, and creating quality specifications are key decision points for ensuring software quality [11].

The quality of the software is significantly impacted by software maintenance. Software that isn't properly maintained might cost more, perform worse, and have fewer features. If not done correctly, maintenance tasks can potentially cause faults and problems, further lowering the quality of the product. Additionally, since badly maintained software can be challenging to use and comprehend, software maintenance can have a significant influence on user happiness. To guarantee that software quality is maintained over time, it is crucial to provide software maintenance operations top priority and financial support [12].

The authors point out that modifications to hardware, software, and user requirements necessitate regular software maintenance. They counter that maintenance may potentially add flaws and lower the caliber of software. The number of changes made, their frequency, their kinds, and their spacing between changes are among the stability indicators the authors provide for gauging the effect of maintenance on software quality. Additionally, they go through how crucial communication and documentation are to preserving software quality when performing maintenance tasks. The article emphasizes the necessity for rigorous management and oversight of software maintenance to prevent a detrimental influence on software quality in its entirety [13]. The authors stress the value of software maintenance since it affects the dependability, maintainability, and usability of the software. The article highlights the difficulties that software maintenance encounters as a result of evolving technology and client demands. To guarantee that software quality is not compromised, the authors also stress the necessity for a wellorganized and disciplined approach to software maintenance. Overall, the article offers a thorough analysis of how software maintenance affects software quality as well as the difficulties and problems that must be resolved in order to guarantee it[14].

Software maintenance is a crucial component of software development that may have a big influence on the caliber of the product. Software that is not adequately maintained can become glitchy, unreliable, and challenging to operate. User annoyance, lost productivity, and in certain cases, even possible safety issues might result from this. Contrarily, well maintained software may continue to be reliable and effective over time, increasing user happiness and improving overall quality. The study "Software Maintenance and Software Quality: A Review" by S. Rathore, S. Gupta, and A. Bhatnagar is one that looks at how software maintenance affects software quality. The study reviews the available research on the subject and offers an overview of the various forms of software maintenance and their effects on software quality. The authors draw the conclusion that proactive maintenance techniques can raise the caliber of software and lower the chance of flaws and other problems [15]. In order to keep the software functioning and up to date throughout its lifespan, software maintenance is a crucial stage in the software development life cycle that encompasses tasks like bug patching, updates, and additions (Juergens, 2016).[16]. Software maintenance, however, may be time-consuming, expensive, and difficult owing to a variety of variables, including developing needs, legacy systems, changing technologies, and problems with documentation (www.journalcra.com, n.d.) [17]. Planning and management must be done properly in order to overcome these obstacles.

In the dynamic field of software maintenance, ongoing research and innovation are needed to solve new problems and provide efficient maintenance methods (Capilla et al., 2011) [18].

To close the gap between research and practice, industry and academics must work together. Practitioners are essential in recognizing the actual difficulties and problems that arise in software maintenance initiatives in the real world. Their knowledge and experiences can help researchers and aid in the creation of successful maintenance plans (Stojanov, n.d.) [19].

According to Yusop and Ibrahim (2011) [20], the major goals of software maintenance are to increase program functionality, raise software quality, fix bugs and defects, improve performance, and guarantee compliance with evolving standards and requirements. For software systems to operate well over the long term, certain goals must be met.

.

III. METHODOLOGY

As the initial step, software maintenance was chosen as the research area for the paper's investigation. Under the umbrella of the software maintenance research area, the title "Investigating the impact of software maintenance activities on software quality" is included.

This comprises the process of upgrading and altering software programs to fix bugs, enhance performance, and satisfy shifting user needs.

The team looked at over 20 studies on software maintenance in order to acquire pertinent research articles for the review. The study subject of examining the effect of software maintenance operations on software quality was identified from this preliminary screening.

Two organizations, Axiata Digital Labs and IFS were chosen for interviews in order to obtain further insight into the real-world use of software maintenance. In order to get both firms' agents' opinions on software maintenance operations and their effects on software, the team created a questionnaire to lead the talks.

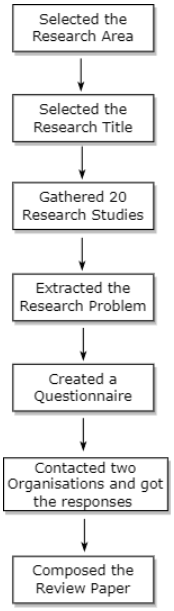


Figure 1: Research Methodology followed for the Study.

The research issue that can be derived from the chosen research subject is "How to minimize software maintenance costs while maintaining software quality," and it has to do with the difficulty of striking a balance between the cost of software maintenance and the requirement to maintain software quality. Although it is a crucial step in the creation of software, software maintenance may be expensive in terms of both time and money. Maintaining software quality is crucial to ensuring that software systems are dependable and fulfill end-user requirements. The challenge in this research is to find solutions to lower maintenance expenses without compromising software quality.

A research problem with software maintenance was finalized after reviewing the comments from the firms. Research publications were chosen for inclusion in the case study using this problem as a guide. After selecting and assessing the findings from the chosen research articles and corporate interviews, the team finally prepared the research paper. The goal of the article was to present a thorough study of how software maintenance operations affect software quality and to pinpoint the most important topics for further study in the domain.

IV.RESULTS

The results from the questionnaire indicate that software maintenance is considered very important in the software development life cycle in the organizations Axiata Digital Labs and IFS R&D International Pvt Ltd. The types of software maintenance activities performed include bug fixing, updates and patches, enhancements and new features, performance optimization, and security updates, all performed at 100% agreement. These maintenance activities are performed as needed, indicating a proactive approach to maintaining software quality.

To measure the impact of maintenance activities on software quality, testing and quality assurance, monitoring and performance metrics, user feedback and satisfaction, and defect and error tracking are used at 100% agreement. This demonstrates a comprehensive approach to ensuring software quality through various means.

To reduce software maintenance costs while maintaining software quality, strategies such as following best coding practices, using automation tools for testing, conducting code reviews, and utilizing version control tools like Git are employed. Allocating resources as needed and adopting a pay-as-you-go method are also utilized to minimize costs.

In terms of software deployment practices, automated deployment, continuous integration and continuous deployment (CI/CD), and cloud-based deployment are used at 100% agreement, while containerization (e.g., Docker) is used at 50% agreement. Software deployment is handled through scheduled maintenance windows, rolling updates, backward compatibility, monitoring and alerting, and automated rollback in case of issues.

In managing software evolution, Agile/Scrum methodology, version control system, requirements documentation and tracking, and user feedback and input are utilized at 100% agreement. Challenges faced in managing software evolution include scope creep, conflicting priorities and stakeholder demands, lack of clear documentation and tracking of requirements, difficulty in managing dependencies and integrations, and limited communication and collaboration among teams. These challenges are addressed through regular communication and coordination among stakeholders, a clearly defined change management process, documentation and tracking of requirements and changes, continuous monitoring and feedback loops, and collaborative tools and technologies for managing changes.

In conclusion, the organizations Axiata Digital Labs and IFS R&D International Pvt Ltd consider software maintenance to be very important in their software development life cycle. Various software maintenance activities are performed, and the impact on software quality is measured through testing, monitoring, user feedback, and defect tracking. Strategies are employed to reduce maintenance costs while maintaining software quality, and software deployment and evolution practices are followed to minimize downtime, ensure backward compatibility, and manage changing requirements. Continuously maintaining the software product is seen as crucial in improving software quality over time.

V. DISCUSSION

One of the key findings from our survey was that changing technologies and evolving requirements were identified as significant challenges in software maintenance. This is consistent with the literature, which highlights the fast-paced nature of the software industry, where technologies and requirements can change rapidly. Legacy systems were also reported as a challenge, which aligns with the research that emphasizes the need to maintain and update older software systems to ensure their continued functionality and compatibility with newer technologies.

Documentation issues were another challenge reported in our survey, and this is supported by the literature, which underscores the importance of adequate documentation for effective software maintenance. Lack of documentation can lead to difficulties in understanding the system's architecture, dependencies, and functionality, which can result in delays and errors during maintenance activities.

Our survey also highlighted the importance of practitioner perspectives in software maintenance. This aligns with the research that emphasizes the practical insights and experiences of practitioners in real-world maintenance projects. Practitioners are often faced with unique challenges and issues that may not be fully captured in academic research. Therefore, collaboration between academia and industry is crucial to bridge the gap between research and practice and develop maintenance strategies that are effective and applicable in real-world scenarios.

Furthermore, the literature emphasizes the need for continuous research and innovation in the field of software maintenance. Emerging issues, such as new technologies, evolving requirements, and changing user expectations, require ongoing research efforts to develop efficient maintenance techniques. Regular updates to maintenance processes, tools, and methodologies are necessary to keep up with the dynamic nature of the software industry and ensure the effectiveness of software maintenance practices. In conclusion, our survey results and the summaries of relevant research highlight the challenges and importance of software maintenance in ensuring the quality and success of software systems. Changing technologies, evolving requirements, legacy systems, and documentation issues are some of the challenges that practitioners face in software maintenance. Practitioner perspectives, collaboration between academia and industry, and continuous research and innovation are crucial in developing effective maintenance strategies. By addressing these challenges and incorporating best practices, organizations can ensure the long-term functionality, performance, and user satisfaction of their software systems.

VI. CONCLUSION & FURTHER WORKS

It is important to not forget software maintenance when developing software. Software quality, usefulness, and compliance with evolving standards may all be improved with proper maintenance. Due to advancing technology, shifting needs, legacy systems, and documentation problems, maintenance can be difficult as well. Therefore, effective administration and planning are essential to overcoming these difficulties.

The existing research indicates that software maintenance efforts, with perfective maintenance having the largest beneficial impact, have a considerable positive influence on software quality. To maintain software quality during maintenance, it's also crucial to have open lines of communication, thorough documentation, and strict management and monitoring.

Future research in software maintenance is needed in a number of areas. One area is the creation of more effective and efficient maintenance methods to handle the problems brought on by expanding needs and changing technology. Examining the use of cutting-edge technologies like artificial intelligence and machine learning for software maintenance is another field.

Research can also look into how software maintenance affects user pleasure, efficiency, and security. Additionally, there is a need for increased industry-academia cooperation to close the knowledge gap in software maintenance.

In general, future research should concentrate on creating novel maintenance methods and approaches that may raise the caliber of software while resolving the difficulties and complexity related to software maintenance.

Best practices for software development can eventually result in a large decrease in the amount of maintenance effort required. Developers may produce more durable and dependable software that needs fewer updates and patches in the long term by adhering to accepted norms and principles for coding, testing, and project management. This not only helps the development team save time and money, but it also enhances user experience by decreasing downtime and raising overall product quality. Best practices must, however, be continually assessed and updated to ensure that they remain applicable and efficient. For optimum efficacy and efficiency, future study can examine how to automate and maximize the use of best practices in software development. the difficulties and complexity of maintaining software.

Best practices for software development can eventually result in a large decrease in the amount of maintenance effort required. Developers may produce more durable and dependable software that needs fewer updates and patches in the long term by adhering to accepted norms and principles for coding, testing, and project management. This not only helps the development team save time and money, but it also enhances user experience by decreasing downtime and raising overall product quality. Best practices must, however, be continually assessed and updated to ensure that they remain applicable and efficient. For optimum efficacy and efficiency, future study can examine how to automate and maximize the use of best practices in software development.

REFERENCES

[1] Grubb, P., & Takang, A. (2010). Software Maintenance and Evolution: A Roadmap. In 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies (pp. 21-26). IEEE.

[2] Riaz, S., Ali, S., & Ahmad, S. (2019). Investigating the Impact of Corrective Maintenance on Software Quality. In 2019 4th International Conference on Computer and Information Sciences (ICCIS) (pp. 1-6). IEEE.

[3]. Zhang, X., & Li, Z. (2015). A Study of the Impact of Software Maintenance on Software Quality. Journal of Software Engineering and Applications, 8(1), 1-8

[4] Martínez-Fernández, S., García-García, J., Piattini, M., & Calero, C. (2016). The Impact of Software Maintenance on Software Quality: A Case Study. Journal of Software: Evolution and Process, 28(9), 733-753.

[5] Ushakova, T., & Kashevnik, A. (2019). Empirical Study of the Effect of Software Maintenance on Software Quality. In

International Conference on Intelligent Data Engineering and Automated Learning (pp. 205-214). Springer.

[6] Kumar, A., Parveen, S., & Kumar, N. (2019). An empirical investigation of the impact of software maintenance on software quality. Journal of King Saud University-Computer and Information Sciences, 31(2), 233-239

[7] Saliu, M. A., & Oladele, T. O. (2020). Impact of Software Maintenance on Software Quality: A Systematic Review. Journal of Software Engineering and Applications, 13(10), 504-517.

[11] Engineering and Emerging Technologies (ICEET) (pp. 1-6). IEEE.

[12] Boehm, B. W., Brown, J. R., & Lipow, M. (1978). Quantitative evaluation of software quality. In Proceedings of the 2nd international conference on Software engineering (pp. 592-605). IEEE Computer Society Press.

[13] Lientz, B.P. and Swanson, E.B., "Problems in Application Software Maintenance," Communications of the ACM, vol. 26, no. 4, pp. 274-287, 1983.

[14] Yau, S. S., & Collofello, J. S. (1990). Some stability measures for software maintenance. IEEE Transactions on Software Engineering, 16(10), 1109-1124. doi: 10.1109/32.60334.

[14] Gupta, A., & Sharma, S. (2016). Software Maintenance: Challenges and Issues. St. Xavier's College, Jaipur - 302001, India. Email: shretasharma@hotmail.com.

[15] Rathore, S., Gupta, S., & Bhatnagar, A. (2016). Software Maintenance and Software Quality: A Review. International Journal of Computer Applications, 135(5), pp. 1-5. Available at: https://www.ijcaonline.org/archives/volume135/number5/24155-2016915967 (Accessed: 22 April 2023).

[16] Juergens, E. (2016) Software maintenance research that is empirically valid and useful in practice, De Gruyter. De Gruyter Oldenbourg. Available at: https://www.degruyter.com/document/doi/10.1515/itit-2016-0014/html (Accessed: April 25, 2023)

[17] www.journalcra.com. (n.d.). Software maintenance | International Journal of Current Research. [online] Available at: https://www.journalcra.com/article/softwaremaintenance [Accessed 25 Apr. 2023].

[18] Capilla, R., Dueñas, J.C. and Ferenc, R. (2011). A retrospective view of software maintenance and reengineering research - a selection of papers fromEuropean Conference on Software Maintenance and Reengineering 2010. Journal of Software: Evolution and Process, 25(6), pp.569–574. doi:https://doi.org/10.1002/smr.548

[19] Stojanov, Z. (n.d.). Software maintenance improvement in small software companies: Reflections on experiences. [online] Available at: https://ceur-ws.org/Vol2913/paper14.pdf.

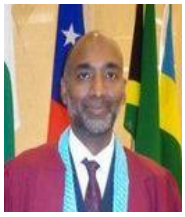
[20] Yusop, O.M. and Ibrahim, S. (2011). Software Maintenance Testing Approaches to Support Test Case Changes – A Review. Communications in Computer and Information Science, pp.33–42. doi:https://doi.org/10.1007/978-3-642-22027-2_4.

[8] Rathore, A. S., & Jaiswal, S. K. (2020). Impact of Software Maintenance on Software Quality: A Review of Empirical Studies. Journal of Computer Science and Applications, 8(1), 1-9.

[9] Bhagat, R., Deshmukh, S., & Phatak, S. (2019). An empirical study of the impact of software maintenance on software quality. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-6). IEEE.

[10] Mubeen, S., Azam, F., & Rehman, M. (2019). Investigating the impact of software maintenance on software quality: a case study. In 2019 International Conference on

AUTHOR BIOGRAPHIES



Nalaka Samantha Lankasena is a Senior Lecturer at the Department of Information and Communication Technology, Faculty of Technology, University of Sri Jayewardenepura ·

VP Pamunuwa is a Software Engineering undergraduate at the Department of Computer Science, Faculty of Computing, KDU.

DP Deraniyagala is a Software Engineering undergraduate at the Department of Computer Science, Faculty of Computing, KDU.

RDAV Thennakoon is a Software Engineering undergraduate at the Department of Computer Science, Faculty of Computing, KDU.

VTB Kulasekara[1]is a Software Engineering undergraduate at the Department of Computer Science, Faculty of Computing, KDU.