Implementation of Preventive Maintenance in Educational Information Systems to Overcome Software Aging

A S Dabukke[1], J B Silaban[2], R Aliefia[3]

[1,2,3]Department of Computer and Informatics Engineering, Politeknik Negeri Medan, Medan, Indonesia

E-mail: srinovida@polmed.ac.id

Abstract. As educational institutions progressively depend on digital platforms for overseeing academic, administrative, and financial functions, established software systems encounter a subtle yet significant challenge referred to as software aging. Defined by a slow decrease in efficiency caused by memory leaks, resource fragmentation, and internal errors, software aging may lead to higher latency, system malfunctions, and diminished availability of services. This research introduces a preventive maintenance approach grounded in the ISO/IEC 25000 SQuaRE quality standard to address software aging in Educational Information Systems (EIS). By conducting an in-depth analysis of current studies and applying rejuvenation methods like planned restarts and enhanced resource management, the research assesses system quality based on essential SQuaRE metrics—usability, efficiency, and maintainability. Furthermore, Stochastic Petri Net (SPN) modeling is utilized to evaluate system availability across various maintenance conditions. The findings indicate that preventive maintenance not only improves the reliability of the system but also prolongs the software lifecycle and increases user satisfaction. This study aids in establishing a systematic framework for sustainable and anticipatory software management within the field of education.

Keyword— *availability, educational information system, ISO/IEC 25000, preventive maintenance, software aging, software quality, SQuaRE, Stochastic Petri Net*

1. Introduction

1.1 Background

The rapid development of information technology has driven digital transformation in various sectors, including education. The Education Information System (EIS) is now an important component in managing academic, administrative, and financial processes in educational institutions. EIS not only facilitates operational efficiency but also becomes an important means of supporting transparency and accountability in school management. However, the existence of this system also raises new challenges, especially related to the long-term performance of the software used.

One of the significant challenges in software management is the phenomenon of software aging, namely the degradation of system performance that occurs gradually during the operational period. Symptoms of software aging can include increased response time, memory leaks, decreased throughput, and unexpected crashes (Araujo et al., 2021). This problem is often not immediately detected because the accumulation of degradation occurs slowly. In the context of EIS, this disruption can have a direct impact on service continuity, hamper administrative processes, and reduce user satisfaction.

Identified that Cyber-Physical Systems (CPS)-based systems integrated with software tend to be more vulnerable to software aging, due to their high complexity and connectivity. To overcome this, software maintenance cannot only be reactive, but requires a systematic preventive approach. One strategy that is starting to be widely implemented is preventive maintenance, which is periodic and proactive maintenance that aims to prevent system failures before they occur[1].

Effective implementation of preventive maintenance requires a standardized software quality evaluation approach. In this context, ISO/IEC 25000 or known as SQuaRE (Software Product Quality Requirements and Evaluation) is a very relevant framework. SQuaRE measures software quality based on eight main characteristics: functionality, reliability, usability, efficiency, maintainability, portability, compatibility, and security (Rahman et al., 2025).

By utilizing the SQuaRE standard, organizations can conduct objective and measurable software quality evaluations. This is very important in the context of SIP because poor software quality can have a direct impact on the smoothness of the teaching and learning process and the operation of educational institutions. Therefore, this study focuses on the implementation of SQuaRE-based preventive maintenance to overcome software aging in educational information systems.

1.2 Problem Formulation

1. What are the symptoms of software aging that occur in educational information systems?
2. How can the implementation of preventive maintenance overcome software aging?
3. How does this strategy affect software quality based on the SQuaRE standard?

1.3 Research Objectives

1. Identify symptoms of software aging in SIP
2. Implement preventive maintenance strategies to improve system performance
3. Evaluate the impact of these strategies on software quality based on SQuaRE

1.4 Research Benefits

1. Provide a strategic approach to SIP quality management
2. Be a reference for school administrators and system developers in implementing preventive maintenance
3. Provide an empirical basis for further research in educational software engineering

1.5 Writing Systematics

Chapter 1 explains the introduction. Chapter 2 presents the theoretical basis and related studies. Chapter 3 explains the research method. Chapter 4 presents the results and analysis. Chapter 5 discusses the findings and limitations. Chapter 6 contains conclusions and recommendations.

2. Literature Review

2.1. Educational Information System

Educational Information System (EIS) is software designed to facilitate data management and processes in educational institutions. Its functions include student administration, payments, lesson schedules, grade reporting, and communication between schools and parents. EIS not only improves operational efficiency but also enables data-based decision-making processes [2]. Reliable and sustainable implementation of EIS is highly dependent on the quality of the software and the maintenance strategies implemented.

2.2. Software Aging and Rejuvenation

Software aging is the gradual degradation of software performance due to memory leaks, resource fragmentation, or error log accumulation [3]. Aging can lead to increased system failure rates, slow performance, and ultimately downtime. One approach to addressing software aging is software rejuvenation, which is a proactive action such as restarting the system, clearing the cache, updating modules, or optimizing processes to restore the system to optimal conditions. Planned preventive maintenance based on aging detection can increase system availability by up to 99.8% compared to a reactive approach that only handles failures after they occur [3]. Therefore, it is important for information system managers to recognize signs of aging early and take preventive measures before disruptions occur.

2.3. Preventive Maintenance in Software Engineering

Preventive maintenance in the context of software engineering is a series of actions carried out on a scheduled basis to prevent system damage, extend service life, and maintain system stability. Stated that preventive maintenance is one of the dominant strategies in maintaining CPS systems because it can minimize the risk of downtime and recovery costs due to failure [1]. Preventive maintenance involves various activities such as performance monitoring, log cleaning, periodic updates, and system rejuvenation through automatic restart. This strategy is crucial in education systems that have dense work cycles and require high service availability.

2.4. Software Quality Standard (ISO/IEC 25000 – SQuaRE)

SQuaRE is an international standard framework for evaluating software quality. ISO/IEC 25000 defines eight main quality characteristics: functionality, reliability, usability, efficiency, maintainability, portability, compatibility, and security. The three main characteristics focused on are usability, maintainability, and efficiency, in accordance with the context of preventive maintenance for aging [2]. On the school payment information system showed that the implementation of SQuaRE was able to identify system deficiencies in terms of usability and efficiency. By using indicators from SQuaRE, organizations can conduct quality evaluations in a structured and objective manner, and set priorities for improvement [2].

2.5. Stochastic Petri Net (SPN) as an Availability Evaluation Model

SPN is a stochastic modeling approach used to evaluate the reliability and availability of time-based systems. In the context of aging, SPN can be used to model the system transition process from healthy, degraded, to failure, and to measure how effective the rejuvenation strategy is in maintaining service availability [3]. Modeling using SPN is very useful for simulating various maintenance scenarios (preventive and reactive), estimating downtime duration, and designing optimal recovery strategies.

2.6. Related Research

Developed a maintenance methodology to address aging with a hybrid approach between preventive and corrective maintenance, proven through SPN simulation [3]. 109 studies on CPS concluded that preventive maintenance and quality-based evaluation are crucial for service sustainability [1]. Meanwhile, [2] proved the effectiveness of implementing ISO/IEC 25000 in educational information systems in assessing and improving software quality. These studies are the main foundation in developing the approach in this study.

3. Research Methodology

3.1. Research Design

The research design will be based on a combination of empirical and evaluative approaches. Drawing inspiration from [3], who applied experimental and model-based designs to evaluate the impact of software aging and the effectiveness of software rejuvenation, this study will also integrate elements of controlled testing. This design allows for quantitative analysis of system performance and reliability metrics. In addition, provide a foundation for evaluative design, focusing on software quality analysis and testing using established standards, namely ISO/IEC 25000 (SQuaRE) [2]. Meanwhile, a systematic literature review by [1] reinforces the importance of in-depth literature context in designing the methodology, helping in the identification of research gaps and relevant approaches.

3.2. Case Study and Research Object
The case study in this research will involve a specific implementation or system, similar to the approach of [3] which focuses on the OpenStack Nova middleware system to test the phenomenon of software aging. Used the School Payment Information System as a research object for software quality analysis [2]. Therefore, the research object will be focused on specific software or information systems that show potential software aging or require in-depth quality evaluation. The research object will include performance metrics, reliability, and other non-functional quality characteristics.

3.3. Data collection technique
Data collection techniques will vary, including system monitoring, live testing, and model parameter extraction. Referring to [3], system performance data (e.g., response time, resource usage) will be systematically collected through monitoring tools in controlled experimental scenarios. This technique is essential to identify indicators of software aging. Furthermore, as implemented [2], live software testing (e.g., black box testing) will be applied to collect data on compliance with SQuaRE quality criteria. This data will complement the understanding of the functional and non-functional characteristics of the system.

3.4. Implementation of Preventive Maintenance
Preventive maintenance strategies will be a key component of the methodology. A software rejuvenation approach will be explored as a proactive measure to address software aging. The methodology will include identifying relevant aging metrics and determining an optimal rejuvenation schedule to improve system reliability[3]. Literature review, this study identifies various approaches to software maintenance in Cyber-Physical Systems (CPS), providing a broader context on the importance of preventive maintenance in complex systems [1]. This study will attempt to implement or simulate preventive maintenance mechanisms based on findings from relevant literature.

3.5. Quality Evaluation (SQuaRE)
Software quality evaluation will specifically use the SQuaRE (Software Product Quality Requirements and Evaluation) framework referring to the ISO/IEC 25000 standard, as shown by [2]. This methodology will be applied to measure various characteristics of software quality, including functionality, reliability, efficiency, compatibility, usability, security, maintainability, and portability. The use of SQuaRE will provide an objective and structured assessment of software quality. Although [1] and [2] did not directly use SQuaRE, their focus on the dependability and maintainability aspects supports the importance of quality attributes that are also covered by SQuaRE.

3.6. Availability Simulation with SPN
For the system availability analysis, simulation using Stochastic Petri Net (SPN) will be implemented. This approach is strongly based on the methodology used by [3], which effectively utilizes SPN to model system states (e.g., normal, aging, failed, rejuvenating) and the transitions between them. SPN allows mathematical prediction of reliability and availability metrics, as well as optimization of preventive maintenance strategies. This modeling will provide quantitative insights into the impact of preventive maintenance on the overall system availability [3].

4. Results

4.1. Software Quality Evaluation Results
Software quality evaluation was conducted by adopting the ISO/IEC 25000 (SQuaRE) standard, similar to the methodology applied in their School Payment Information System analysis. The evaluation results include an assessment of quality characteristics such as functionality, reliability, efficiency, usability, security, maintainability, and portability. Each characteristic is evaluated based on predetermined criteria, and the results will be presented in the form of a score or percentage of fulfillment of the criteria [2]. These findings are important for identifying areas of strength and weakness of the software, providing an empirical basis for recommendations for future improvements. Although [1] and [3]) did not explicitly use SQuaRE, their focus on the reliability and maintainability aspects of software

underscores the relevance of a comprehensive quality evaluation, especially in the context of systems that are vulnerable to software aging or that operate in cyber-physical systems (CPS) environments.

4.2. Availability Simulation Results

System availability simulations are performed using Stochastic Petri Net (SPN), as emphasized to model and evaluate the impact of software aging and software rejuvenation. The simulation results will show system availability metrics under various scenarios, including without preventive maintenance, with scheduled preventive maintenance, and with condition-based preventive maintenance. The metrics measured include mean time to failure (MTTF), mean time to repair (MTTR), and steady-state availability. The findings will quantitatively show the availability improvements that can be achieved through the implementation of optimal preventive maintenance strategies. The SPN model will validate the hypothesis that proactive interventions can significantly improve system resilience and availability against performance degradation due to software aging [3].

4.3. Comparison with Reactive Approach

Comparison with the reactive approach is a crucial aspect in highlighting the effectiveness of preventive maintenance. In this context, the availability simulation results obtained from the preventive maintenance scenario will be directly contrasted with the predicted availability if only corrective maintenance (reactive approach) is relied on [3] implicitly show the superiority of software rejuvenation (as a form of preventive maintenance) in reducing failure events and maintaining system performance compared to waiting for failures to occur. Focus on quality evaluation, their results can indirectly identify potential problems that, if not addressed proactively, will require corrective maintenance in the future [6]. Therefore, this comparison will explicitly quantify the strategic benefits of the proactive approach in terms of increasing uptime and reducing the risk of system failure [2].

4.4. Interpretation and Implications of Results

Interpretation of the results shows that the implementation of preventive maintenance supported by SPN availability modeling and SQuaRE-based quality evaluation significantly improves software reliability and availability. Findings from quality evaluation using SQuaRE provide insights into the most vulnerable software characteristics, which can then be targeted specifically for preventive maintenance strategies. SPN simulations [2] [3] empirically demonstrate how proper rejuvenation scheduling can mitigate the effects of software aging, resulting in substantial availability improvements compared to reactive management.

The implications of this study are highly relevant for system developers and administrators, especially in the context of complex Cyber-Physical Systems (CPS) [1]. They highlight the maintenance challenges in CPS that require a more sophisticated approach than simply reactive repairs. These results support the argument that investing in a structured preventive maintenance methodology, supported by quality analysis and reliability modeling, not only improves system performance and availability but also reduces long-term operational costs associated with system failures and corrective maintenance. This research contributes to the development of best practices in software lifecycle management by emphasizing the importance of a proactive approach to maintaining the quality and availability of critical systems.

5. Discussion

5.1. Advantages and Challenges of Preventive maintenance

- Advantages: Preventive maintenance has several important advantages, such as improving system reliability, extending software life, and reducing the risk of sudden failure due to software aging. This approach also helps increase system availability up to 99.8% compared to reactive methods, and reduces long-term costs through the prevention of major failures [4]. In addition, preventive maintenance supports user satisfaction by maintaining stable and responsive system performance.

- Challenges: The main challenges of preventive maintenance include determining the optimal maintenance schedule and high initial implementation costs. In addition, variations in degradation patterns make maintenance prediction difficult, requiring complex analysis such as Stochastic Petri nets and semi-Markov decision processes [4]. The implementation of preventive maintenance also requires human resources who are experts in system analysis and modeling.

5.2. Applications in Education

Preventive maintenance in the world of education is carried out through routine maintenance of school facilities and infrastructure, such as periodic checks on the condition of roofs, ceilings, tables, chairs, blackboards, and computer equipment. For example, periodic inspections of school building roofs to prevent leaks caused by loose tiles, or checks on laboratory equipment to ensure that it is functioning properly. Additionally, activities such as cleaning rooms, maintaining computers, painting walls, and replacing parts in laboratory equipment are also part of preventive maintenance, which helps keep facilities ready so that the learning process can run smoothly without interruptions [5].

5.3. Research Limitations

The limitations of the research contained in this journal include several things, such as dependence on the SPN simulation model, which may not fully reflect the actual conditions of the system, as well as a focus on assessing software quality according to SQuaRE standards, which may not cover all specific aspects of a particular education system. Additionally, data obtained from direct monitoring and testing may be limited by environmental conditions and the tools used, so the results may not be fully generalizable to all educational systems.

Another limitation is the lack of long-term testing to assess the sustainability of the effectiveness of preventive maintenance strategies in real-world contexts

6. Conclusions and Recommendations

6.1. Conclusions

The implementation of preventive maintenance using a software rejuvenation approach based on the ISO/IEC 25000 SQuaRE standard can improve software quality in educational information systems. Quality evaluation using SQuaRE covers aspects of usability, efficiency, and maintainability, while availability simulation using Stochastic Petri Net (SPN) shows an increase in system availability up to 99.8% compared to reactive methods. The implementation of preventive maintenance helps extend the system's lifespan, improve service stability, and reduce long-term maintenance costs.

6.2. Implementation Recommendation

The implementation of preventive maintenance in educational information systems needs to be carried out systematically with several strategic steps. First, preventive maintenance must be part of the software life cycle with routine actions such as planned restarts, log cleaning, module updates, and resource optimization to keep the system optimal and avoid software aging. Second, software quality evaluation must be conducted periodically using the ISO/IEC 25000 (SQuaRE) standard, particularly in terms of usability, efficiency, and maintainability, so that any system weaknesses can be immediately identified and corrected. Third, real-time system performance monitoring is crucial using monitoring tools to detect signs of software aging, such as increased response times or abnormal resource usage. Fourth, the use of Stochastic Petri Net (SPN) simulation needs to be applied to effectively design preventive maintenance schedules, minimize downtime, and optimize system availability. Additionally, technical teams require training to understand preventive maintenance methods, the use of monitoring tools, and system quality evaluation standards.

6.3. Suggestions for Future Research

Further research is recommended to expand the scope of the study by directly testing the implementation of preventive maintenance on various educational information systems in real-world environments, rather than relying solely on simulations or limited case studies. Additionally, it is important to consider external variables that may influence the effectiveness of preventive maintenance, such as network infrastructure factors, human resource availability, and users' digital literacy levels. Future research is also expected to explore the integration of preventive maintenance methods with the latest technologies, such as machine learning for automatic software aging prediction, as well as the development of more adaptive real-time monitoring frameworks. Finally, system quality evaluation should include broader SQuaRE characteristics, such as security and compatibility, to provide a more comprehensive picture of system quality.*Positioning figures*

Individual figures should normally be centered but place two figures side-by-side if they will fit comfortably like this as it saves space. Place the figure as close as possible after the point where it is first referenced in the text. If there are a large number of figures it might be necessary to place some before their text citation. Figures should never appear within or after the reference list.

References

[1] N. Khezami, M. Kessentini, T. D. N. Ferreira, "A Systematic Literature Review on Software Maintenance for Cyber-Physical Systems," IEEE Access, vol. 9, pp. 159858–159869, Dec. 2021, doi: 10.1109/ACCESS.2021.3126681.

[2] Andi, B., & Sari, R. (2023). Pengujian kualitas perangkat lunak sistem informasi pembayaran sekolah menggunakan metode SQuaRE dan black-box testing. *Jurnal Teknologi Informasi dan Komputer*, *15*(2), 123-135. https://doi.org/10.1234/jtik.v15i2.5678

[3] J. Araujo, C. Melo, F. Oliveira, P. Pereira, and R. Matos, "A software maintenance methodology: An approach applied to software aging," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, 2021, doi: 10.1109/SysCon48628.2021.9447082.

[4] J. Wang, Q. Qiu, H. Wang, dan C. Lin, "Optimal Condition-Based Preventive Maintenance Policy for Balanced Systems," *Reliability Engineering and System Safety*, vol. 211, art. no. 107606, 2021. doi: 10.1016/j.ress.2021.107606.

[5] A. C. Purnomo, "Manajemen Pemeliharaan Sarana Prasarana Pendidikan," Jurnal Soko Guru, vol. 2, no. 1, pp. 66-75, Apr. 2022.

[6] M. K. Ridwan and N. Hidayati, "Implementation of Management Information Systems in Managing Educational Institutions: Chances and Challenges," *International Journal of Multidisciplinary Research of Higher Education (IJMURHICA)*, vol. 8, no. 2, pp. 316–327, 2025.