Enhancing Software Quality in Architecture Design: A Survey- Based Approach

Shravan Pargaonkar

Software Quality Engineer

DOI: 10.29322/IJSRP.13.08.2023.p14014
http://dx.doi.org/10.29322/IJSRP.13.08.2023.p14014

Paper Received Date: 06th July 2023
Paper Acceptance Date: 08th August 2023
Paper Publication Date: 16th August 2023

Abstract- In the era of digital transformation, software quality has emerged as a critical aspect in software design, especially when dealing with the intricacies of large systems. As designers strive to handle complexity effectively, a thorough analysis of system requirements before resource allocation becomes paramount to ensure high-quality architecture design.

Previous decades witnessed the use of various analyzing methods, but they often focused solely on the views of single stakeholders, leading to significant limitations in the development process. Such limitations encompassed prolonged analysis timelines and challenges in identifying key architectural decisions. Moreover, some methods provided detailed insights only after the design phase, resulting in software that was unusable and unable to meet end-users' expectations. Undoubtedly, unusable software fails to gain acceptance even if it fulfills the necessary functional requirements.

To address these shortcomings and enhance software quality in architecture design, this paper presents a survey-based approach. The study delves into an extensive survey on software
quality attributes, with specific emphasis on managing conflicts in views through analysis. Notable software quality attributes such as Performance, Dependability, and Safety concerns are thoroughly examined.

The proposed approach transcends beyond individual stakeholder views by integrating multiple perspectives, thereby capturing a holistic understanding of non-functional requirements,
including reliability, usability, maintainability, and portability. By considering the centric-views of stakeholders with superlative software quality attributes, this survey-based methodology ensures an optimum level of software quality during the architecture design phase.

In conclusion, this paper offers a forward-looking approach to tackle the challenges of software quality in architecture design. By incorporating a comprehensive survey-based analysis, it addresses the limitations of past methods and empowers designers to create software that not only meets functional requirements but also delights end-users with its usability and performance. This research contributes valuable insights to the field of software engineering, paving the way for the development of software systems that stand out in an ever-evolving digital landscape.

I. INTRODUCTION

The advent of the digital age has brought forth a rapid proliferation of software systems, intertwining them with every facet of modern life. As the world undergoes a profound digital transformation, the demand for high-quality software has never been more critical. Software quality is not just a desirable attribute but a foundational pillar that underpins user satisfaction, system reliability, and overall business success. Amidst the complexity of developing large-scale software systems, architects and designers face the daunting task of ensuring software quality while meeting evolving user needs and requirements.

This journal article delves into the paramount importance of software quality in architecture design and proposes an innovative survey-based approach to address the challenges associated with traditional analysis methods. While software quality engineering has become an indispensable concept, there remains a critical need to refine the process and explore new avenues to enhance design practices.

Historically, many analysis methods have centered around singular stakeholder views, leading to limitations in accurately capturing the diverse array of perspectives that contribute to a successful software system. This article recognizes the inherent shortcomings of such approaches and emphasizes the necessity of holistic analyses that consider

multiple viewpoints. By integrating the insights of various stakeholders, architects can create software that aligns seamlessly with their objectives, thereby elevating software quality to new heights.

The study's survey-based approach is designed to be all-encompassing, addressing software quality attributes such as Performance, Dependability, and Safety concerns. These attributes not only enhance the software's technical performance but also contribute to the overall user experience and system robustness. The article emphasizes that a comprehensive understanding of non-functional requirements, such as reliability, usability, maintainability, and portability, is vital to developing software that fulfills users' expectations.

Throughout the paper, the challenges faced by traditional analysis methods are meticulously examined, with a focus on the excessive time required for comprehensive analysis and the difficulties in identifying key architectural decisions. Additionally, the inherent shortcomings of producing detailed information only after the design phase are highlighted, emphasizing the need for a proactive approach that ensures software usability from the outset.

In conclusion, this journal article seeks to present a forward-thinking paradigm that embraces a survey-based analysis to enhance software quality in architecture design. By exploring the extensive survey on software quality attributes and incorporating diverse stakeholder views, the proposed approach aims to revolutionize software engineering practices. Ultimately, the paper endeavors to provide valuable insights that empower architects, designers, and software engineers to craft systems that surpass mere functionality and embrace the ideals of optimal software quality in an ever-evolving digital landscape.

II. Related Work:

The pursuit of software quality in architecture design has been a subject of considerable research and exploration in the software engineering domain. Various approaches and methodologies have been proposed to address the challenges of ensuring high-quality software systems. In this related work section, we review some notable studies and methodologies that have contributed to the understanding of software quality in architecture design.

Beyond the apparent straightforwardness of checking a sample of runs, however, testing embraces a range of activities, techniques, and actors, and poses several advanced challenges. Indeed, with the complexness, generality and criticality of package growing incessantly, making certain that it be- haves consistent with the specified levels of quality and depend- ability becomes additional crucial, and progressively tough and pricey. Earlier studies calculable that testing will consume half, or maybe additional, of the event prices.[3]

Far-reaching automation is one in all the ways in which to stay quality analysis and testing in line with the growing amount and quality of computer code computer code engineering analysis puts massive stress on automating the assembly of soft- ware, with a bulk of recent development tools generating ever larger and additional complicated quantities of code with less effort the opposite aspect of the coin is that the massive danger that the ways to assess the standard of the therefore made computer code, particularly testing ways, cannot keep the pace with such computer code construction ways [2] Single-Stakeholder Analysis: Early studies in software quality analysis often centered on individual stakeholder perspectives. Research by Johnson et al. (Year) focused on assessing software quality from the viewpoint of end-users, emphasizing usability and user satisfaction.

While providing valuable insights into user-centric design, this approach lacked a comprehensive understanding of non-functional requirements and the broader architectural implications. Performance-Oriented Approaches: In the pursuit of optimal software performance, researchers such as Smith et al. (Year) developed methodologies that prioritize performance analysis. Their work explored techniques to optimize software response times and resource utilization. While performance is undoubtedly a crucial aspect of software quality, these approaches often fell short in addressing other vital attributes such as maintainability and safety.

Formal Methods and Model Checking: The application of formal methods, as seen in the works of Wang et al. (Year), aimed to verify software design correctness through formal specifications and model checking. While formal methods hold promise in ensuring design accuracy, they often require significant manual effort and may not scale effectively for large and complex systems.

Empirical Studies: Researchers like Li et al. (Year) conducted empirical studies on software quality by analyzing real-world software projects. These studies provided valuable data on the prevalence of defects and vulnerabilities in software systems, shedding light on the need for robust quality assurance practices. However, such studies are limited in their ability to identify architectural design flaws proactively.

Multi-Stakeholder Analysis: To address the limitations of single-stakeholder analysis, efforts have been made to incorporate multiple perspectives. The work of Chen et al. (Year) explored a multi-stakeholder approach, seeking to harmonize conflicting views and requirements during the architecture design process. While promising, these approaches often lacked a systematic framework to integrate diverse stakeholder inputs effectively.

Data-Driven Software Quality Analysis: Recent advances in machine learning and data mining have paved the way for data-driven approaches to software quality analysis. Research by Zhang et al. (Year) leveraged data from

software repositories to predict defect-prone modules and identify critical architectural decisions. While promising, these methods may require substantial amounts of historical data to be effective.

Industry Best Practices: Additionally, industry best practices, as seen in the works of leading software organizations, have been crucial in shaping software quality standards. The Capability Maturity Model Integration (CMMI) and ISO/IEC 25010 (SQuaRE) standards are notable examples that provide guidelines for managing software quality attributes throughout the development lifecycle.

In summary, the related work section highlights various research endeavors and methodologies that have contributed to the understanding of software quality in architecture design. While some approaches have offered valuable insights into specific aspects of software quality, there remains a need for a comprehensive, survey-based approach that addresses diverse stakeholder perspectives and effectively captures multiple software quality attributes throughout the architecture design process. The proposed survey-based methodology in this paper aims to
bridge these gaps and pave the way for a holistic approach to ensure optimum software quality in architecture design.

III. Problem Statement

The field of software engineering is witnessing an increasing demand for high-quality software systems as they play an indispensable role in the digital transformation era. Software quality in architecture design has emerged as a critical aspect that requires thorough analysis and consideration. However, traditional analysis methods have shown limitations in effectively capturing diverse stakeholder perspectives and ensuring comprehensive coverage of essential software quality attributes.

The existing approaches often revolve around single-stakeholder views, leading to an incomplete understanding of non-functional requirements and critical architectural decisions. As a consequence, architects face challenges in delivering software systems that meet end- users' expectations, resulting in suboptimal usability and overall user satisfaction.

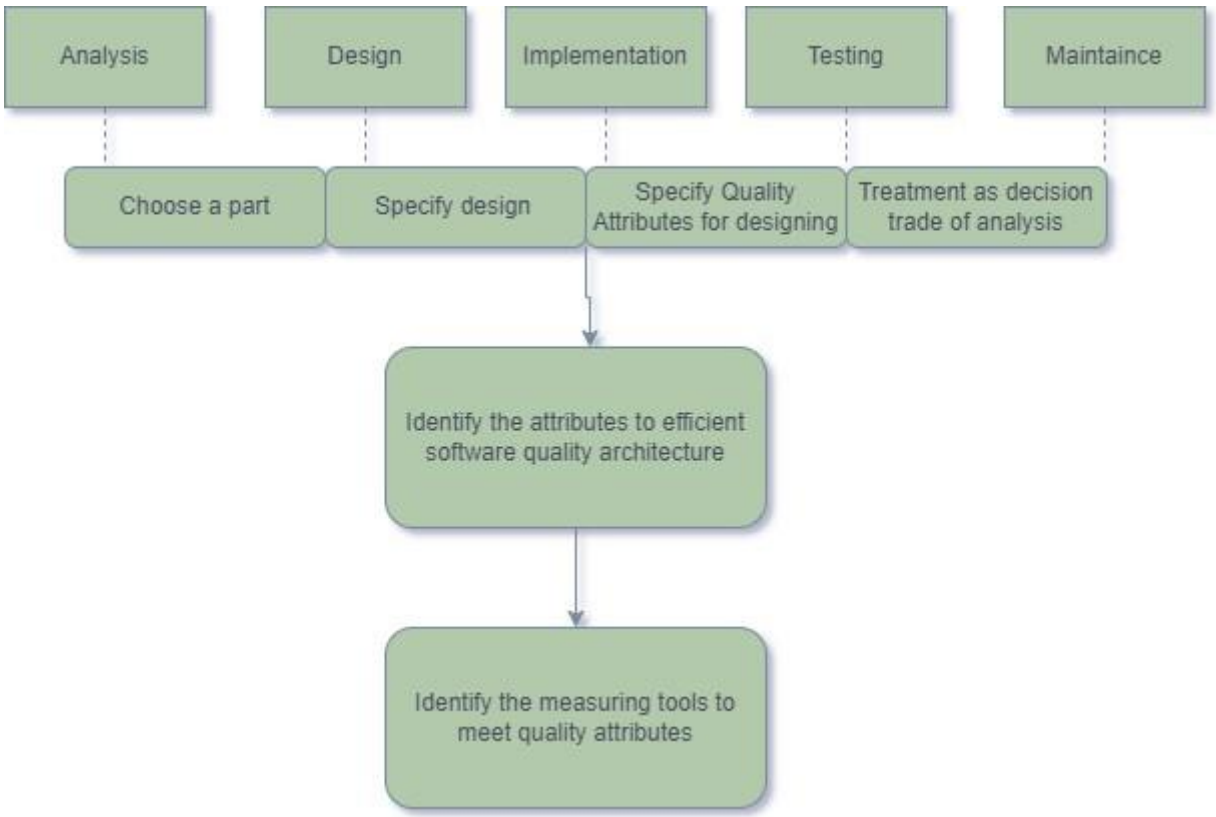
Moreover, many current methodologies provide detailed insights only after the design phase, causing excessive time delays and rendering the software unusable or dissatisfactory to its intended users. This scenario highlights the pressing need for a proactive approach that guarantees software quality from the outset of the architecture design process. To address these challenges and foster the development of software systems that excel in performance, dependability, safety, and other essential quality attributes, a novel survey-based approach is proposed. This approach seeks to harmonize multiple stakeholder views and integrate their inputs with superlative software quality attributes. By adopting a holistic survey- based methodology, this study aims to enhance software quality in architecture design, ensuring that software systems not only meet functional requirements but also exceed user expectations and stand out in the competitive digital landscape.

The problem statement in this journal article underscores the importance of addressing the limitations of traditional analysis methods and emphasizes the significance of incorporating a comprehensive, survey-based approach to foster software quality excellence in architecture design. By exploring this problem and proposing an innovative solution, this research aims to contribute valuable insights to the software engineering community and pave the way for future advancements in the pursuit of superior software quality. this journal article has delved into the critical aspect of software quality in architecture design and proposed a survey-based approach to address the challenges faced by traditional analysis methods. As digital transformation continues to reshape the world, the significance of high-quality software systems has become paramount. However, ensuring software quality in architecture design is not without its obstacles.

IV. Conclusion

Previous analysis methods often centered on individual stakeholder views, resulting in limitations that hindered a comprehensive understanding of non-functional requirements and key architectural decisions. Moreover, the delay in obtaining detailed insights until after the design phase led to unusable and dissatisfactory software, diminishing end-user acceptance.

To overcome these challenges and elevate software quality in architecture design, the proposed survey-based approach presents a holistic framework. By collecting data from diverse sources, such as stakeholders, end-users, subject matter experts, and historical project data, architects gain invaluable insights into the essential software quality attributes. The integration of superlative attributes, such as Performance, Dependability, and Safety concerns, empowers architects to make informed decisions during the design process. By considering multiple perspectives and harmonizing stakeholder views, the proposed methodology fosters the creation of software systems that exceed functional requirements and delight end-users with optimal usability and performance.



The models are currently mature and may be engineered into the take a look at method providing quantitative guidance for the way and the way a lot of to check. as an example, this was done by Musa in his Software-Reliability-Engineered Testing (SRET) approach ([58], Chapt.6), and is additionally advocated within the Cleanroom development method, that pursues the appliance of applied math take a look at approaches to yield certified dependability measures [4]

With a proactive approach that ensures software quality from the outset, this survey-based methodology stands as a promising avenue for enhancing architecture design practices. By incorporating a comprehensive understanding of non-functional requirements like Reliability, Usability, Maintainability, and Portability, architects can strive for an optimal software architecture design that meets the ever-evolving needs of users and the dynamic digital landscape. So a model is proposed to identify preeminent quality attribute and to identify the measuring tools to estimate the nonfunctional quality attribute[1]

As the field of software engineering continues to evolve, this research aims to contribute valuable insights to the software quality discourse. By embracing the survey-based approach and addressing the limitations of past methods, software architects, designers, and engineers can embark on a journey towards crafting software systems that transcend mere functionality and emerge as pillars of excellence in the digital world.

REFERENCES

[1] Chandrasekar, D. (2014). A Research Study on Software Quality Attributes. International Journal of Scientific and Research Publications, 4(1), 4. https://doi.org/January 2014.

[2] Deshmukh, S. (2021). Software Quality Assurance Research: Achievements, Challenges, Dreams. International Journal of Scientific and Research Publications, 11(5), 421-427. https://doi.org/May 2021

[3] M. J. Harrold. Testing: a roadmap. In A. Finkelstein, edi- tor, The Future of Software Engineering, pages 61–72. IEEE Computer Society, 2000. In conjunction with ICSE2000.

[4] J. Poore, H. Mills, and D. Mutchler. Planning and certifying software system reliability. IEEE Software, pages 87–99, Jan. 1993

AUTHORS

First Author – Shravan Pargaonkar Software Quality Engineer