International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 03 | March – 2026 DOI:10.55041/ISJEM05940
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Integrating Quality Assurance into Web Application Development for Enhancing Software Reliability and Defect Prevention

Kanojiya Rohankumar[1] , Mr. Girrajsinh Lavendrasinh Puvar[2]
[1]Student, Department of Computer Science & Engineering, Parul University, Vadodara, Gujarat, India
[2]Assistant. Professor, Department of Computer Science & Engineering, Parul University, Vadodara, Gujarat, India

ABSTRACT - Quality assurance is becoming more and more important to the success of current-day web applications, as the failure of a system impacts user confidence and business performance. Traditional software development methodology separates quality assurance from development, as it generally waits until the end of the software development lifecycle to conduct quality assurance. The result is that many defects are not identified until after the completion of the project and thus add to the maintenance costs of the project. The research presented in this paper proposes an alternative software development methodology where quality assurance practices are incorporated into each of the phases of the software development lifecycle rather than being treated as a standalone phase. The research was based on a case study of a Laravel-based web application. The results from the case study demonstrated that by integrating quality assurance into the software development process, the number of defects found during testing was reduced, regression stability improved, and traceability of requirements improved. By integrating quality assurance into the software development process, software quality is improved, and software defects that develop after deployment are minimized.

KEYWORDS: Quality Assurance, Software Reliability, Defect Prevention, SDLC, Web Development

I. INTRODUCTION

Web apps are structured in a distributed, complex, environment. Design relies upon quality assurance (QA) to function correctly. A minor issue with improper input confirmation or handling may create a cascade effect resulting in excessive breakdown of the product, which creates an unstable application or complete product failure.

Quality assurance (QA) has traditionally been treated as the final step or the latter half of the development process. One writes the code first, executes a test later (after the code is written). The delay in performing these steps is a financial burden because bugs are often found very late in the SDLC, not only requiring additional work to correct the error, but also increasing the risk of implementing a bug, and placing unnecessary time constraints on the SDLC.

Studies show a significant quantity of resources are associated with correcting a defect if the defect is discovered early in development when compared to correcting non-detected defects late during development [14]. With the rapid growth of modern development practices, the approach towards QA has changed. QA is no longer a separate activity at a remote phase of the process; it has become a facet of the overall workflow. QA activities will be conducted as an integral part of a project's life cycle from requirements confirmation to design evaluation to code testing and subsequently as a continuous activity as opposed to merely a checkpoint.

This paper explores if using an integrated approach to QA throughout the life cycle will have a beneficial impact on the systems' reliability. Utilizing an analysis of the number of defects identified in a specific instance of the integrated approach within a case study on the reliability of a functional web application.

II. LITERATURE SURVEY

Reliability, maintainability, and defect prevention have been related to software quality for a long time; however, one definition of reliability is the likelihood that an item will not fail during some period of time when used under prescribed conditions [3][14].

A theme that emerges from numerous pieces of prior research is the need to find defects early. According to Boehm, costs to fix a defect increase as development continues, so it is more efficient to prevent than to correct defects.

The majority of defect prevention models emphasize the use of structured inspections, validation techniques, and process discipline to decrease the amount of defects injected into software [6]. Additionally, multiple studies on the software development life cycle (SDLC) show that embedding QA into the various phases of development increases system stability and reduces the amount of defect leakage from development into production [4].

Agile and DevOps methodologies expand on these concepts through continuous integration (CI) and continuous testing (CT; thus, defects are discovered virtually immediately upon their introduction into the code base) [10]. Yet many small and academic projects are still using traditional post-development testing techniques. This disconnect between what is theorized and what is actually done has resulted in the proliferation of unstable software systems and high rates of defects. The intent of this study is to close this gap by implementing and assessing an integrated QA approach to software development in a practical setting.

III. METHODOLOGY / APPROACH

The methodology of this research follows case studies to investigate the effectiveness of incorporating Quality Assurance (QA) in the software development process.

A. System Environment

The experiment was conducted using the following technology stack on a web-based management system:

- Backend: PHP
- Framework: Laravel
- Database: MySQL
- Testing approach: Manual structured testing

B. Research Phases

The research was conducted in three phases:

1. Baseline Development (BD)

The web-based management system was developed initially without any QA integration. QA was applied after all development was completed.

2. QA-Integrated Development

All phases of development included QA activities from the beginning to end, including requirements verification, design review, code inspections and continuous testing.

Figures 1 and 2 show the Software Testing Life Cycle (STLC) and how QA activities are integrated into every phase of the System Development Life Cycle (SDLC) respectively.

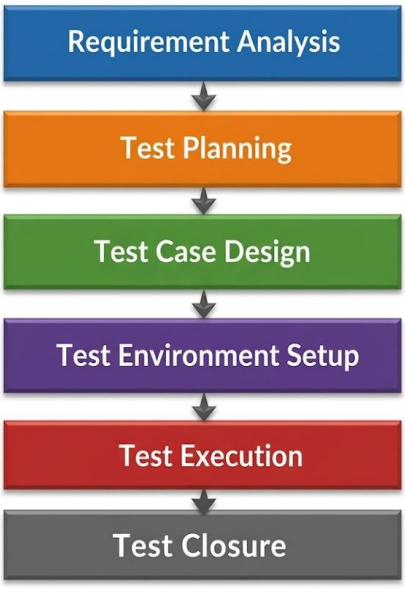

Fig 1. Software Testing Life Cycle (STLC)

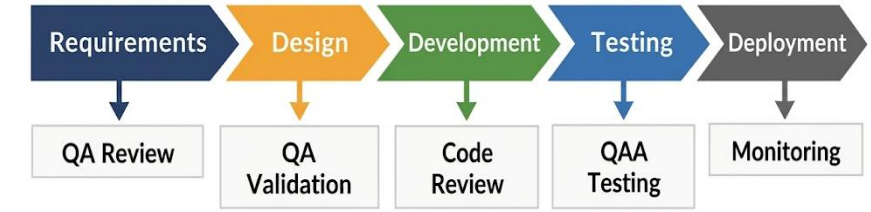

Fig 2. QA-Integrated Software Development Life Cycle

3. Comparative Evaluation

Once both development processes had been completed, a comparative analysis was performed on both sets of performance metrics to determine the amount of improvement in using QA as part of the development process. This study follows a case study-based methodology to evaluate the effectiveness of QA integration within the development process.

C. Integration Methods for Quality Assurance

The Quality Assurance implementation methods used included:

- Validating requirements and creating requirement traceability matrices
- Using code reviews and peer-to-peer inspections
- Validating inputs and performing boundary testing
- Verifying exception handling processes
- Testing sessions/workflows

The method used in the QA implementation is depicted in Figure 3, Requirement Traceability Matrix, and more specifically, in Figure 4, Defensive Coding Validation Architecture.

Fig 3. Requirement Traceability Mapping

Fig 4. Defensive Coding Validation Architecture

D. Evaluation Metric

Quantitative methods were used to measure the performance of the QA integration methodology using the following:

- Defect Density
- Regression Failure Rate
- Code Rewrite Percentage
- Defects After Deployment
- Requirements Traceability Coverage

These evaluation metrics will be a basis for analyzing software reliability and identifying opportunities to improve quality.

IV. RESULTS & DISCUSSION

The inclusion of Quality Assurance (QA) into the Development Cycle led to measurable improvements in many different areas of quality.

A. Defect Reduction

Overall defects have decreased following the implementation of QA. Early detection and identification in the Development Cycle have aided in finding defects that would normally have gone undetected until deployment.

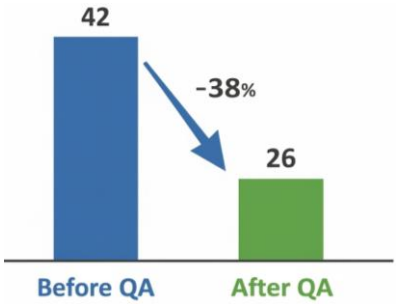

Fig 5. Defect Reduction After QA Integration

B. Regression Stability

Continuous validation and structure testing have decreased the number of regression failures experienced. This means that when there are changes made to the current functionality, the changes do not negatively affect any current capability.

C. Code Quality Improvement

Using code inspection and peer review processes has resulted in an increase in the developability of code and a reduction in the amount of rework performed by developers. By identifying logical errors during the Development Cycle through code inspection and peer reviews have provided developers the opportunity to focus on correcting Logical mistakes before they occur.

D. Requirement Traceability

The requirement traceability matrix has provided a consistent and structured means of ensuring that all requirements are traceable to a corresponding Test case. Through the use of the requirement traceability matrix, the Development Team achieved 100% traceability of all requirements to corresponding test cases, thus providing a means by which the organization will be able to demonstrate increased reliability of the system.

E. Comparison of Results

Metric	Before QA	After QA	Improvement
Total Defects	42	26	-38%
Critical Defects	7	3	-57%
Regression Failures	11	5	-56%
Code Rework	32%	14%	Reduced
Post-Deployment Defects	8	2	-75%

The results of the QA integration process demonstrate that there has been an improvement in system performance, reduction in defects and improvement in the reliability of the system as compared with that of the QA integration.
Figure 6 depicts the increasing costs associated with correcting defects during various stages of the Software Development Life Cycle and Figure 7 depicts the Improvements in reliability over time.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 03 | March – 2026 DOI:10.55041/ISJEM05940
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

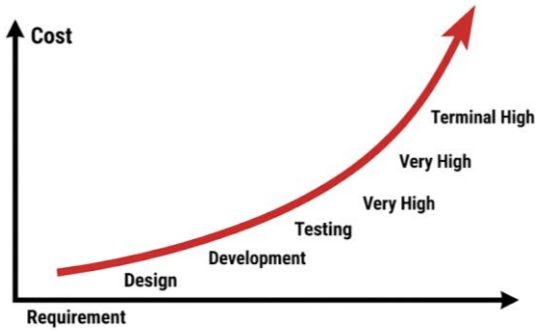

Fig 6. Cost of Defect Fixing Across SDLC

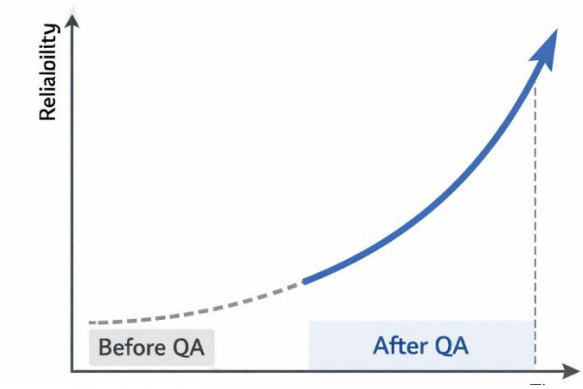

Fig 7. Reliability Improvement Trend

F. Discussion

The results suggest that there is now a move from reactive defect discovery to a proactive means of preventing defects. QA integration allows for constant vigilance and confirmation as the product is created, as opposed to identifying problems only when the project is completed. Thus, using this type of approach (process) leads to lower costs for the company to develop a product and increases customer satisfaction by creating a more stable and reliable product

V. CONCLUSION

The results of this study show that integrating quality assurance into the software development process has been found to improve software reliability and help reduce the number of defects found in software products. The quality assurance integrated approach results in the ability to identify and address problems early in the software development process; as a result, there is a reduction in the amount of rework required and an overall increase in the quality of the system being developed.

The findings of this research highlight that software quality cannot be obtained from testing alone, and that it must be developed as part of the software development process through ongoing systematic validation and continuous monitoring. As next steps, other areas to explore may be the use of automated testing tools, CI/CD pipelines, and advanced defect prediction methods to improve software quality and the efficiency of the software development process.

REFERENCES

[1] R. K. Ramesh and A. Reddy, "Integrating Software Assurance into the Software Development Life Cycle," 2012.
[2] A. Alenezi, "Software Quality Models: Exploratory Review," 2023.
[3] A. Gupta et al., "Software Reliability and Quality Assurance Challenges in CPS Security," International Journal of Computer Science and Security, 2021.
[4] S. Kumar and P. Singh, "Importance of Testing and QA in SDLC Models," International Journal of Soft Computing and Engineering, 2012.
[5] M. Sharma, "Ensuring Software Quality Through Effective QA Testing," 2023.
[6] M. I. A. et al., "Defects Prediction and Prevention Approaches for Quality Software Development," International Journal of Advanced Computer Science, 2018.
[7] H. Zhang et al., "Software Process Improvement Based on Defect Prevention," 2021.
[8] W. Humphrey, "Experiences with Defect Prevention," 1995.
[9] V. Suma and T. R. G. Nair, "Effective Defect Prevention Approach," 2010.
[10] M. Hilton et al., "Effects of Continuous Integration on Software Development," 2021.
[11] J. Smith and R. Brown, "Defect Prediction Models in Software Engineering," 2019.
[12] T. Johnson, "The Crucial Role of Inspection in Software QA," 2018.
[13] P. Verma and R. Shah, "Improving Software Quality Through Defect Management," CIIT Research Journal, 2015.
[14] B. Boehm, "Testing in the SDLC: Now or Later," 1999.
[15] L. Briand et al., "Web Testing for Reliability Improvement," 2005.
[16] K. El-Emam, "Leveraging Defect Prediction Metrics," 2012.
[17] R. Patel and S. Mehta, "Analysis of QA Practices Across SDLC," Journal of Emerging Technologies and Innovative Research (JETIR), 2022.
[18] M. Schwaber, "How Scrum Adds Value to Achieving Software Quality," 2022.
[19] IEEE, "International Symposium on Software Reliability Engineering (ISSRE)," IEEE Proceedings.