ABSTRACT

In the context of modern software development approaches, the rapid delivery of functional software through incremental and iterative releases often introduces challenges in ensuring reliability and managing associated costs. Frequent requirement changes, overlapping development and testing phases, and evolving customer expectations necessitate robust models that can accurately predict software faults and guide optimal release planning based on reliability and associated cost. This research aims to enhance software reliability assessment and cost-aware decision-making in Agile environments by developing a comprehensive framework that integrates fault prediction, severity, fault correction, and cost modeling across multiple sprints.

The proposed work introduces both parametric and non-parametric reliability models that consider the dynamics of fault detection and correction processes with a focus on fault severity. Faults are classified into minor, major, and critical categories, and their correction patterns are modeled using two-step fault removal processes that reflect real-world practices such as beta testing and post-release debugging. Analytical models capture the fault accumulation and removal behavior over successive releases, while advanced neural architectures, such as Recurrent Neural Networks and Attention-based Bidirectional Gated Recurrent Units, are employed to handle temporal sequences of fault data in multi-sprint scenarios. These data-driven models improve the prediction accuracy and adaptability in Agile contexts.

In addition, a reliability and cost-optimized release planning framework is developed that incorporates multiple cost components, including testing, debugging, beta testing, and warranty costs. This framework evaluates the trade-off between reliability improvement and cost minimization across multiple sprints, enabling the identification of optimal release time. Severity-based cost coefficients and remaining fault handling across sprints are also integrated to reflect realistic software development and maintenance scenarios.

The findings of this research demonstrate the effectiveness of incorporating severity-aware modeling, two-step fault removal, and hybrid modeling approaches for release time determination. The developed models provide practical insights for software development teams and decision-makers, supporting the delivery of reliable software within constrained budgets and schedules in Agile settings.

Keywords

Software Reliability, Agile Development, Software Reliability Growth Model, Fault Detection, Fault Correction, Fault Severity, Optimal Release Time.