Chapter 7

An Investigation on Quality Perspective of Software Functional Artifacts

Vimaladevi M.
Pondicherry Engineering College, Puducherry, India

Zayaraz G.
Pondicherry Engineering College, Puducherry, India

ABSTRACT

Software engineering process and practices paramount the crisis of cost, quality, and schedule constraints in developing software products. This chapter surveys the quality improvement techniques for the two fundamental artifacts of software product development, namely the architecture design and the source code. The information from top level research databases are compiled and an overall picture of quality enhancement in current software trends during the design, development, and maintenance phases are presented. This helps both the software developers and the quality analysts to gain understanding of the current state of the art for quality improvement of design and source code and the usage of various practices. The results indicate the need for more realistic, precise, automated technique for architectural quality analysis. The complex nature of the current software products requires the system developed to be beyond robust and resilient and building intelligent software that is anti-fragile and self-adaptive is favored. Innovative proposals that reduce the cost and time are invited.

INTRODUCTION

Software Engineering has emerged legitimately in developing high quality software products right from its inception. The discipline of Software Engineering evolved over the past 70 years shaping its key activities providing a framework to the stakeholders to structure, plan, develop and control the software development process. The origin of software engineering dates back to 1950s, where the initial crisis was productivity and now it has evolved to quality. Software quality assessment and improvement is a vast area of research and many techniques and processes are proposed for quality improvement in various stages of the software life cycle. Any quality assurance technique strives to achieve zero errors post release. In spite of all these constant and effective techniques, there are still some failures in the software that makes the software difficult to survive. Irrespective of the type of software and the technology used in development, all software products face the challenges in incorporating high quality within the cost and time constraints. There are multiple definitions of Quality such as conformance to requirements, satisfying customer needs, achieving zero defects, etc. Software applications are becoming more complex day by day and it is difficult to maintain code quality that make the Quality-Cost balance a challenging task.

The study reported in this work will lay a foundation of quality needs and various strategies available that the stakeholders may select to build high quality software products considering the design architectures and the source code. It is mandatory to evaluate a software quality and the quality assessment has to be performed in parallel with the software development. Every industry is now computerized and is used in critical areas where quality becomes a key factor to ensure successful business and human safety. Software is becoming more and more complex and it is mandatory to select, apply and evaluate relevant techniques and processes to keep the risks low. Evaluation has to be done in order to understand a software product. Understanding involves testing of software whether it is easy to use, hard to modify, can be integrated with other programs, etc.

There exists a vast set of literature that discuss about software quality. They focus on a specific phase of a software development or adhere to certain techniques and tools. This work in contrast to the existing literature reviews; the quality aspects taking the two major artifacts of the software development are studied. They are the Architectural Design and the Source Code. All the Software Engineering principles, process models, quality frameworks, testing tools and techniques are aimed in fulfilling the user requirements and thus achieved desired quality. Even though the software development process generates different artifacts at different phases, all that is used to increase the quality of the developed source code. The quality of the source code is directly related to the design choices made during the

23 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/an-investigation-on-quality-perspective-of-software-functional-artifacts/235765

Related Content

AI for Health-Related Data Modeling: DCN Application Analysis
Na Cheng (2022). *International Journal of Information System Modeling and Design (pp. 1-11).*
www.irma-international.org/article/ai-for-health-related-data-modeling/300780

Maintaining Transactional Integrity in Long Running Workflow Services: A Policy-Driven Framework
Stephan Reiff-Marganiecand Manar S. Ali (2013). *Service-Driven Approaches to Architecture and Enterprise Integration (pp. 135-164).*
www.irma-international.org/chapter/maintaining-transactional-integrity-long-running/77948

Requirements Engineering Process Improvement and Related Models
Badariah Solemon, Shamsul Sahibuddinand Abdul Azim Abd Ghani (2012). *Software Process Improvement and Management: Approaches and Tools for Practical Development (pp. 18-33).*
www.irma-international.org/chapter/requirements-engineering-process-improvement-related/61208

Keystroke-Based Biometric Authentication in Mobile Applications
Fatima Kabli, Sad houari Nawaland Matoug Afaf (2022). *International Journal of Software Innovation (pp. 1-16).*
www.irma-international.org/article/keystroke-based-biometric-authentication-in-mobile-applications/303574

Critical Success Factors for an Effective Security Risk Management Program
Humayun Zafar (2022). *International Journal of Systems and Software Security and Protection (pp. 1-26).*
www.irma-international.org/article/critical-success-factors-for-an-effective-security-risk-management-program/315765