Chapter 35
Quality, Improvement and Measurements in High Risk Software

Edgardo Palza Vargas
University of Ottawa, Canada

ABSTRACT

We aim to improve quality in software engineering projects by introducing Verification and Validation (V&V) best practices in terms of process, artifacts, and quality performance indicators. We present a Quality Measurement Management Tool (QMT) in order to support quality activities and process. This tool is based on a measurement meta-model repository for collecting, storing, analyzing and reporting measurement data. It is important to note that the proposed QMT supports the IEEE Standard 1012 for Software Verification and Validation (management, acquisition, supply, development, operation, and maintenance), as well as measurement information needs to Capability Maturity Model Integration (CMMI) processes and products requirements. Our repository is generic, flexible, and integrated, supporting a dynamic measurement system. It was originally designed to support Ericsson Research Canada's business information needs.

INTRODUCTION

Since software has become an important component of critical systems (e.g., in aeronautics/aerospace, power plants, medical devices, chemical plants, automobiles, military weapons, etc.) the impact of software on systems safety has been demanding greater attention in organizations dedicated to the production of this high critical software. In the same vein today's software is becoming increasingly more complex: heterogeneous composition on a diversity of platforms, distributed execution, complexity in calculation algorithms, multiplicity of contractors with diverse development methodologies, etc. The result of such complexity is increased risk and higher costs in software projects.

The type of software that, directly or indirectly, ensures the safety of human life or significant financial investments is referred to as high risk Software or safety-critical software. This type of software is required to meet very high levels of safety and reliability and to meet demanding quality standards. Therefore, the related development process must be tightly managed, because of the very high level of quality required. Indeed, many accidents caused by deficient quality of critical software have been reported in the software engineering literature (Levenson, 1993), Ariane 5 (Lions, 1996), etc.

In this chapter we propose a Quality Measurement Management (QMT) in order to support quality activities and process. This tool is based on a measurement meta-model repository (Abran & Palza, 2003; Palza, 2010; Palza, Abran, & Fuhrman, 2003). It is important to note that the proposed QMT supports the IEEE Standard for Software Verification and Validation (IEEE Std 1012, 2004), as well as CMMI best practices (CMMI Product Team, 2006). The software processes described in IEEE Std. 1012 include: management, acquisition, supply, development, operation, and maintenance. This IEEE standard is also recommended for use in software-intensive projects: for instance, the NASA IV&V Facility's "Program Manager Handbook" (NASA Software IV&V Facility, 2000) makes clear the usefulness of the IEEE Std 1012 and (IEEE Std 1059, 1993) for the planning and execution of V&V activities in their projects.

The effectiveness of V&V depends on the timeliness of the development processes and on the quality of the deliverables. In this proposal, we discuss how the QMT can help to support measurements activities for Management V&V processes in organizations. Our approach is described in terms of activities, processes and tasks recommended in IEEE 1012. This should facilitate measurements in terms of:

- Software Verification and Validation Plan (SVVP).
- Baseline change assessment.
- Management Review of V&V.
- Management and Technical Review Support.
- Interface with Organizational and Supporting Process.

Moreover, this chapter also reports how QMR tool supports a dynamic measurement system with CMMI requirements in a changing and dynamic environment, such as that of Ericsson Research Canada. The QMR tool was expected to meet further criteria and constraints for a better fit to the Ericsson context:

- Improvements to the quality of the software engineering measures.
- Design of a coherent and consistent model of enterprise performance evaluation.
- An integrated and generic multidimensional measurement platform.
- Individual and team performance measures aligned with organizational goals.
- Ability to allow managers to extract value from the vast amounts of data and information in the organization.

Quality in High Risk Software

Software V&V is a set of activities aimed at attaining software quality during the development life cycle. Although there are several approaches (or models) to planning the life cycle for software development, there are clearly certain disciplines that exist in phases within any life cycle. These disciplines include requirements engineering, analysis, design, implementation and testing. At each step along the development life cycle, mistakes can be made, which in turn can affect the quality of the final software product.

Related Content

Machine Learning-Based Electricity Load Forecast for the Agriculture Sector
Megha Sharma, Namita Mittal, Anukram Mishraand Arun Gupta (2023). *International Journal of Software Innovation (pp. 1-21).*
www.irma-international.org/article/machine-learning-based-electricity-load-forecast-for-the-agriculture-sector/315735

A Multi Agent Based Approach for Critical Components Identification and Testing
D. Jeya Malaand R. Iswarya (2014). *International Journal of Systems and Service-Oriented Engineering (pp. 21-38).*
www.irma-international.org/article/a-multi-agent-based-approach-for-critical-components-identification-and-testing/104652

Case Study of Software Reviews
Yuk Kuen Wong (2006). *Modern Software Review: Techniques and Technologies (pp. 253-267).*
www.irma-international.org/chapter/case-study-software-reviews/26907

Speaking of Software: Case Studies in Software Communication
Ann Brady (2009). *Software Engineering: Effective Teaching and Learning Approaches and Practices (pp. 75-97).*
www.irma-international.org/chapter/speaking-software-case-studies-software/29594

Design Churn as Predictor of Vulnerabilities?
Aram Hovsepyan, Riccardo Scandariato, Maximilian Steffand Wouter Joosen (2014). *International Journal of Secure Software Engineering (pp. 16-31).*
www.irma-international.org/article/design-churn-as-predictor-of-vulnerabilities/118146