

Chapter 3.2

Information Communication Technology Tools for Software Review and Verification

Yuk Kuen Wong
Griffith University, Australia

INTRODUCTION

While information communication technology (ICT) can be considered a well-established discipline, software development projects are still prone to failure. Even if a software project is not classified as a failure, the general level of software quality leaves room for much improvement. It has been suggested that one of the most prevalent and costly mistakes made in software projects today is deferring the activity of detecting and correcting software problems until the end of the project. Hence, the cost of rework in the later stages of a project can be greater than 100 times the project costs. About 80% of avoidable rework comes from 20% of defects. As a result, techniques such as software review for improving software quality are important.

Software review (inspection/verification) was originally introduced by Fagan (1976). The review process essentially includes six major steps:

1. **Planning:** Organize and prepare the software review, typically for preparing the review materials and review procedure, forming review team and scheduling review meeting, selecting review participants, and assigning roles.
2. **Overview:** Author explains overall scope and the purpose of the review.
3. **Individual preparation:** Individual reviewers analyze and review the software artefact.
4. **Group review meeting:** Find errors, sometimes also called “logging meeting.” Review teams correct and the reader summarizes the work.
5. **Rework:** Defect correction, which involves the author in resolving problems by reviewing, revising, and correcting the identified defect or by decreasing the existence of errors of the software artefact.
6. **Follow-up:** Validate the correction quality and decide if re-inspection is required

Since Fagan (1976) introduced software review and verification as an important technique to assure the quality of software projects, researchers have investigated ways to improve software review performance. It has been suggested that ICT software review tools are one of the important elements to the support software review process. This article overviews ICT tools to assist software review and verification during this software review process. The overall objective of this article is to identify various ICT tools that support software review and verification. This includes a discussion of the importance of software quality and identifies ICT tools for effective software reviews and verification.

ICT TOOLS FOR SOFTWARE REVIEW AND VERIFICATION

Intelligent Code Inspection in a C Language Environment (ICICLE)

The ICICLE (Intelligent Code Inspection in a C Language Environment) is the first published software review tool, which was developed at Bellcore (Brothers, Sembugamoorthy, & Muller, 1990). The ICICLE tool is designed to support code review and assists reviewers in both individual preparation and group meetings. ICICLE provides a synchronous communication support to group meetings. It has been argued that traditional code review meeting is manually documented (i.e., using paper and pen to record defects detected). This documentation procedure is very time consuming, tedious, and could be inconsistent recording (Brothers et al., 1990). One of the aims of this tool helps software reviewers to find obvious defects. Brothers and his team (1990) suggested that ICICLE provide several benefits to code review:

- To detect routine sorts of errors, with the goal of freeing the code inspector (reviewer) to

concentrate on verifying the correct implementation of requirement, specifications, and designs.

- To offer various forms of knowledge about the code being inspected (reviewed), including domain and environment knowledge, and information from various forms of analysis such as cross-referencing.
- To allow code inspectors (reviewers) to easily traverse source code in a windowed environment instead of riffling through hard copy from many different files.
- To render the code inspection (review) meeting paperless through a shared window interface which enables the code inspectors (reviewers) to fulfill their roles electronically.

The ICICLE tool consists of two phases in the review process: the individual review and group review meeting. The group review meeting takes in the same location/venue, usually a reviewers' seat at nearby computers. An individual reviewer allows entering comments on each line of code. According to MacDonald, Miller, Brooks, Roper, and Wood (1995), the researcher found that "the computer supported meeting format appeared to cause substantial changes in the dynamics of the code inspection (review) meeting." In other words, the procedures of the code review meeting using ICICLE can enable roles during the group meeting process (Brothers et al., 1990). For example, the additional duty of a moderator is to record statistics relating to coding defects discovered during code review. The reader can direct the attention of the other reviewers to areas of interest in the source code. The scribe's records must be agreed on by the review team. The author should present in the code review meeting and answer the reviewers' question. Any additional reviewers can participate and share meeting discussions.

7 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/information-communication-technology-tools-software/29438

Related Content

Testing E-Learning Websites

Kamaljeet Sandhu (2012). *Advanced Automated Software Testing: Frameworks for Refined Practice* (pp. 182-195).

www.irma-international.org/chapter/testing-learning-websites/62156

Integrated Software Testing Learning Environment for Training Senior-Level Computer Science Students

Daniel Bolanos and Almudena Sierra (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 2029-2045).

www.irma-international.org/chapter/integrated-software-testing-learning-environment/29493

Construction and Implementation of Content-Based National Music Retrieval Model Under Deep Learning

Jing Shi and Lei Liu (2024). *International Journal of Information System Modeling and Design* (pp. 1-17).

www.irma-international.org/article/construction-and-implementation-of-content-based-national-music-retrieval-model-under-deep-learning/343631

Face Biometric Authentication System for ATM Using Deep Learning

D. Jeyamani Latha, V. Yogalakshmi and E. Swathi (2023). *Cyber-Physical Systems and Supporting Technologies for Industrial Automation* (pp. 187-196).

www.irma-international.org/chapter/face-biometric-authentication-system-for-atm-using-deep-learning/328500

Comparing Misuse Case and Mal-Activity Diagrams for Modelling Social Engineering Attacks

Peter Karpati, Guttorm Sindre and Raimundas Matulevicius (2012). *International Journal of Secure Software Engineering* (pp. 54-73).

www.irma-international.org/article/comparing-misuse-case-mal-activity/66408