

# 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 concen-

trate 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.

## Scrutiny

Scrutiny is an online collaborative software review tool, which was developed at Bull HN Information Systems in conjunction with the University of Illinois (Gintell, Houde, & Mckenney, 1993). It is a synchronous meeting review tool. It is one of the early comprehensive collaborative process software review tools (MacDonald et al., 1995). Scrutiny currently supports text documents only.

Scrutiny can be used in the formal review process, and it supports multi-users review but does not support for rules and checklists. It provides a "petri-net based process modeling language" that allows the system to implement alternative software review methods, such as a "shared preparation" phase in which reviewers have access to each other's preliminary findings (Gintell et al., 1993; MacDonald et al., 1995). However, in comparison with the ICICLE, the Scrutiny usage can depart radically from manual software review processes, such as geographically distributed software reviews (MacDonald et al., 1995).

## Collaborate Software Inspection (CSI)

Collaborate software inspection (CSI) was built and used in a case study to compare online distributed computer-mediated software review meetings vs. face-to-face software review meetings at the University of Minnesota (Mashayekhi, Feulner, & Riedl, 1994). As with other software review tools, the CSI provides the similar process characteristics of Humphrey's software review method with hypertext capability. CSI is developed for group review meetings in the (1) same time and place, (2) same time and place, (3) same time and different place, and (4) different time and same place. CSI supports both asynchronous and synchronous activities that include materials distribution, individual preparation/individual review, group review meeting, recording, and reporting (Mashayekhi et al., 1994).

## InspeQ

InspeQ was developed to support the phased software review process (Knight & Myers, 1993). The InspeQ was executed on "Sun 3, Sun 4, and IBM RS/6000 computers running various forms of Unix and the X-window display system and the OSF/Motif widget set" (Knight & Myers, 1993). Although the InspeQ achieves the goals of efficiency and rigor in the phased review process, it is not viewed as essential to the "phased inspection" method (MacDonald et al., 1995). Similar to other software review tools, InspeQ provides numbers of facilities to support software review process. These include work product display (views the documents), checklist display (allows the status of each checklist to be displayed and modified), standard display (review rational and a detailed descriptions), highlight display (helps locating particular aspects of the documents),

5 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/13389](http://www.igi-global.com/chapter/information-communication-technology-tools-software/13389)

## Related Content

---

### Electronic Records Management at a Federally Funded Research and Development Center

Susan M. Hendrickson and Margo E. Young (2014). *Cases on Electronic Records and Resource Management Implementation in Diverse Environments* (pp. 334-350).

[www.irma-international.org/chapter/electronic-records-management-federally-funded/82658](http://www.irma-international.org/chapter/electronic-records-management-federally-funded/82658)

### Factors Affecting Consumer Behavior in Omni-Channel Marketing of the Fitness Industry: Motivation and Attitude Towards O2O Channels

Lung-Yu Li and Long-Yuan Lee (2022). *Journal of Cases on Information Technology* (pp. 1-10).

[www.irma-international.org/article/factors-affecting-consumer-behavior-in-omni-channel-marketing-of-the-fitness-industry/306244](http://www.irma-international.org/article/factors-affecting-consumer-behavior-in-omni-channel-marketing-of-the-fitness-industry/306244)

### RFID and Labor Management Systems Selection in the Logistics Industry

Cheryl A. Tibus and Linda L. Brennan (2010). *Journal of Cases on Information Technology* (pp. 31-49).

[www.irma-international.org/article/rfid-labor-management-systems-selection/40322](http://www.irma-international.org/article/rfid-labor-management-systems-selection/40322)

### The Influence of National and Organizational Cultures on Technology Use: An Exploratory Study within a Multinational Organizational Setting

Zixiu Guo and John D'Ambra (2010). *Information Resources Management: Concepts, Methodologies, Tools and Applications* (pp. 1710-1731).

[www.irma-international.org/chapter/influence-national-organizational-cultures-technology/54566](http://www.irma-international.org/chapter/influence-national-organizational-cultures-technology/54566)

### Factors Influencing the Use of Decision Support Tools of Enterprise Resource Planning Systems

Emad M. Kamhawi (2010). *Global, Social, and Organizational Implications of Emerging Information Resources Management: Concepts and Applications* (pp. 205-224).

[www.irma-international.org/chapter/factors-influencing-use-decision-support/39243](http://www.irma-international.org/chapter/factors-influencing-use-decision-support/39243)