

Books

High Quality Low Cost Software Inspections, Ronald A. Radice, Paradoxicon Publishing, Andover, MA (2002). 479 pp. (ISBN 0-9645913-1-6).

This book by Ronald Radice describes the author's 28 years of experience defining, using, and teaching the software inspection process. He states that his goal is to first focus on the effectiveness and then on the efficiency of inspections. Inspection effectiveness is how to find the most defects as early as possible at controlled and low costs. Inspection efficiency is finding the most defects at an increasingly lower cost to continue ROI (return on investment) improvement for the inspection processes. The book is very complete on the subject of inspection, covering software inspection history, inspection process definition and evolution, inspection economics, management, practical issues, where to start, what to inspect, other possible approaches, and the future. The appendix contains checklists, materials, and forms to help anyone who would like to pursue software inspections.

I believe the book is very readable and should be read by anyone who is interested in learning more about software inspections, including programmers, quality assurance and test personnel, project leaders and managers, inspection champions, Software Engineering Process Groups (SEPGs), and teachers. Readers who are just learning about software inspections, as well as those with inspection experience, will find the book helpful. The book gives an excellent overview of software inspection processes, how to develop a process tailored to your organization or project, and how to describe the business benefits of inspections to all involved. The author is extremely well qualified to write this book, having been significantly involved in the original Fagan inspections and in the leading software process improvement and

quality activities in IBM. Radice also led the use of software inspections at Bull, was director of the Software Process Program in the Software Engineering Institute (SEI) at Carnegie Mellon University, and recently was a principal partner in Software Technology Transition, a company that provides training, consulting services, assessment services, and software engineering method solutions.

The book is organized in four main parts. Background and introduction are the topics in Chapters 1 and 2, the software inspection process in Chapters 3 through 8, inspection economics and management in Chapters 9 through 12, and other approaches and the future in Chapters 13 and 14. I briefly review key highlights that I found in each of these parts.

Background and introduction—Chapters 1 and 2. Chapter 1, "Background," answers the question of why inspections are needed. "Our objective with Inspections is to reduce the Cost of Quality by finding and removing defects earlier and at lower cost." A high-level view is given of the benefits of inspections in the software development process along with answering some key questions, including: Why isn't everyone using inspections? Do good programmers make mistakes? When do we start? I found the historical perspective section that describes the first inspection in the IBM VTAM* (Virtual Telecommunications Access Method) project in 1972 to be particularly interesting, because I was personally involved with the introduction of "Fagan inspections" in 1975 in the System/38* operating system project in the IBM Rochester Development Laboratory.

©Copyright 2003 by International Business Machines Corporation.

Chapter 2, “An Introduction,” defines inspections, examines how they are defined in this book using the ETVX (Entry-Task-Validation-Exit) Process Definition Paradigm, describes where they fit in the software development life cycle, their cost, and how they compare to “walkthrough” and “review.” All are important introductory topics that are well described. The section on effectiveness and efficiency provides clear well-thought-out definitions that are used throughout the book. Chapters 1 and 2 should be read completely by all.

Software inspection process—Chapters 3 through 8. The software inspection process flow is clearly defined in Chapter 3 using a consistent format for each procedure. The format includes an introduction, describes the individual responsible for process, other roles, entry criteria, tasks to be performed, validation/verification, and exit criteria. The exit criteria include the measurements that are important to the procedure. The procedures are complete in reference to the original “Fagan Process,” IEEE Inspection Process definition, and proven process improvements. Importantly, a scissors icon shows suggested tailoring options. This chapter is an excellent starting point for the organization interested in defining a software inspection process tailored to its requirements and structure.

Complete descriptions for the moderator role are provided in Chapter 4, and descriptions are given for the other inspection roles, including producer, recorder, and inspector in Chapter 5, as well as their various activities in the different inspection procedures. Of particular importance in Chapter 5 are discussions about process training, rules of behavior for effective meetings, how to select participants, and inspection psychology. Introducing software inspections in an organization is a major cultural change and requires careful planning and thought for the software inspections to be successful.

Chapter 6, “Inspection Data,” is extremely important. This chapter should be well understood by all those who are implementing and using software inspections. The chapter is very complete with basic measurement definitions and how to use the data to plan, monitor, and improve the inspection process. Basically, size, effort, cost, and time are measured to provide an analysis about inspection defect density, defect distribution, inspection meeting rate, preparation rate, inspection efficiency, error-prone work products, and inspection effectiveness. Because not everything can be inspected and work needs to

be prioritized, in my opinion, the information about error-prone work product analysis is very important for software inspections to be effective in any organization.

Causal analysis and reinspections are covered in Chapters 7 and 8, respectively. Causal analysis is a defect prevention process that tries to identify the probable cause that led to defects in the first place. It can be used with defects from the inspections to provide learning and to prevent defects from happening over and over. It is an important process improvement exercise. Chapter 8 describes when to reinspect. Defect density, large work products, inspection rate, preparation rate, complexity, and change rate are all among the topics that should be considered when making a reinspection decision. A reinspection checklist that will be helpful to a team making this decision is included.

Inspection economics and management—Chapters 9 through 12. Chapter 9, “Economics of Inspections,” is another very important chapter. This chapter covers the costs of inspections and how to determine the return on investment for them. Inspection effectiveness is discussed, as well as the value of inspections in finding defects early in the life cycle, when they can be fixed with significantly less expense than when the defects are found by test or by the customer. Software inspections done properly will save in overall project cost and schedule time primarily by reducing test effort and shortening test schedules.

In Chapters 10, 11, and 12 the topics of managing inspections, practical issues for success, and what to inspect are discussed. Important to inspection success is management commitment, funding, adequate resource, and process management. The author makes an important point that for inspections to be successful an inspection coordinator needs to be appointed to assist in managing the execution of the inspection process. Chapter 11, “Practical Issues for Success,” provides answers to the many day-to-day questions that come up when executing a successful inspection process. Examples of issues are: getting off to a good start, do we really need all those data with inspections, and inspections waste time. Chapter 12 describes work products that can be inspected. They include requirements specifications, architecture definitions, high- and low-level designs, and code. Inspections can be applied to any type of work product, and usually the same process will work with

a change in the checklist to support the work product being inspected.

This section is especially important for management, project leaders, moderators, and the SEPG to manage the effectiveness and operation of a successful inspection process.

Other approaches and the future—Chapters 13 and 14. Chapter 13 describes other defect removal processes—reviews, walkthroughs, and producer self-checking—that are frequently confused with software inspections. They are usually not as effective and often do not provide data to determine their effectiveness. An IEEE checklist is provided listing the characteristics of an inspection process.

A discussion of the future of inspections in Chapter 14 details where the author believes software inspections are headed. Software inspection processes already have had many improvements over the past 25 years, including tools and improved work product format and presentation. The case for selective and solo inspections is described. The last section on aspects of a best case inspection process lists ideas for ongoing inspection process improvement, detailing how inspections have evolved.

Summary. Overall, I recommend this book to anyone interested in starting or improving an inspection process. The book is complete with an inspection process that can be tailored to the needs of an organization. The author has included many examples and tips that will be helpful in making the inspection process of an organization successful. *High Quality Low Cost Software Inspections* is a complete guide to the design and use of successful software inspection processes.

Richard J. Hedger
Quality Software Technologies, Inc.
Rochester, Minnesota

**Trademark or registered trademark of International Business Machines Corporation.

Note—The books reviewed are those the Editor thinks might be of interest to our readers. The reviews express the opinions of the reviewers.