Software Technical Reviews: A Practical Guide

Karl E. Wiegers

December 1, 2000

Table of Contents

Preface (700 words)

Objectives of this Book

The purpose of this book is to help the reader perform several varieties of software technical reviews (STRs) on work products created in his or her organization. It also provides guidance on both the cultural and the technical aspects of implementing an effective STR program in a software organization. The inspection process is emphasized as a formal review technique, but several other review methods are also described, spanning a spectrum of formality. The book's emphasis is on straightforward practical review approaches that any software development organization concerned about quality can apply.

Audience

This book is written for people who realize that the quality of their software products falls short of their goals, people who want to validate or tune up their current technical review practices, and people who want to ship high-quality software on schedule. The target audience includes potential participants in technical reviews, inspection moderators, software project and product managers, quality assurance and testing professionals, and process improvement leaders. Groups who are pursuing Level 3 of the Capability Maturity Model for Software will also find this book valuable. The techniques described are not specific to software: they can be applied to any technical or engineering work product, including design specifications, schematics, and assembly instructions.

Chapter 1. The Quality Challenge (3000 words)

Chapter Objective: *To help the reader understand some basic concepts and issues in software quality and appreciate why building quality into software products is cost effective. To define quality control practices and identify technical reviews as a critical QC practice. To differentiate STRs from other types of review.*

Peeking Over the Shoulder

Nearly every programmer has had the experience of asking a colleague to help find an elusive problem in his code. Often, we are too close to our own work to find errors we've made. A technical review is any activity in which someone other than the author of a work product examines that product with the express intent of finding defects and improvement opportunities. Technical reviews are contrasted with other types of project reviews, such as project status reviews, management reviews, educational or informative reviews, post-project reviews, and brainstorming or problem-solving reviews.

The Cost of Quality

The cost of quality is the price we pay for doing things wrong, not the cost of doing them right [Crosby]. "Quality is free" means that the investment you make in building quality into a product is more than repaid by reducing the costs of late-stage defect correction and reducing the number of problems the customer finds. Managers, developers, and customers are sometimes opposed to reviews because they believe reviews are ineffective or will slow down the project. In reality, reviews don't slow down the project—bugs do. A team that wants to release a quality product will find that STRs are not just cost-effective but also time-effective. To build the case for using technical reviews to find defects shortly after they are injected into the product, literature data [Boehm, Grady] are presented that illustrate the accelerating cost of correcting defects that are found later in—or following—the

development process. Reviews can reduce the cost of quality, reduce work, and hence improve development productivity by shifting defect detection to early stages of product development. However, each project team needs to balance the expected benefits from STRs against its need to release a product very quickly, reaching an appropriate business balance between development speed, feature content, product quality, and long-term maintenance costs. In the Internet world, time-to-market is often viewed as paramount, but the cost of quality on Internet projects includes lost business opportunities because customers don't use poor products more than once. Introducing technical reviews is an important aspect of software process improvement that can reduce your cost of quality and improve the efficiency and effectiveness of a development organization.

Quality Control and Technical Reviews

People make mistakes, both small and large. Whether a defect is major or minor depends on the context. STRs are a set of verification techniques that determine whether a work product satisfies its specification. The software development V-model life cycle is presented to illustrate both testing and technical review activities for verifying the requirements, architecture, detail design, and code. Inspection has been identified as one of the most significant software industry best practices [Brown]. The relationship of manual inspections to other static checking methods and tools is discussed. Strategies and selection guidance for using automated rather than manual methods are presented. For example, automated methods can find even subtle syntax violations or likely problem areas such as uninitialized variables, but they aren't likely to detect logic errors in the implementation. Similarly, a word processor's spell-checker can catch misspelled words, but it is unable to spot factual errors. Several reasons why reviews are superior to testing in many cases are described, and the types of errors best found by testing and review are identified. The goal of technical reviews is not just to catch errors, but also to learn how to create better work products. I've learned something from every review in which I've participated as either an author or a reviewer.

What Can Be Reviewed

Although many people associate reviews with source code, virtually any software development or project management work product can benefit from peer review. Other products that are ripe for review include requirements specifications and models, business process models, charter documents, architecture descriptions, design documents and models, program and system documentation, test cases and procedures, all kinds of project plans, and user documentation. Few organizations can afford to formally review all of their work products, so you should select the components to be reviewed based on the overall impact to the project of having residual defects go undetected. The highest leverage from STRs comes from their application to early-stage work products, such as requirements specifications. Appropriate stages for reviews of products created through incremental development life cycles are suggested. Non-software technical work products, such as circuit board specifications or designs, also can be reviewed.

Chapter 2. A Little Help from Your Friends (7000 words)

Chapter Objective: *To describe characteristics of reviews and the cultural aspects of STRs. Describe what makes reviews succeed in an organization.*

Scratch Each Other's Back

In a healthy, quality-focused culture, team members willingly "hire" their peers to help them improve their work products. Asking your colleagues to point out defects in your work is a learned, not instinctive, behavior. Jerry Weinberg's concept of egoless programming is summarized and updated. The time you spend reviewing a team member's work is repaid

when he or other team members review your own work, thereby helping you improve the quality of your work products and increase your productivity. An effective review program provides benefits for the author (finding errors), the reviewers (learning), the project (faster development), the development organization (reduced maintenance effort), and the customer (higher customer satisfaction. When you've internalized the benefits of technical reviews, you will feel uncomfortable unless another pair of eyes has examined any significant work product you created.

Reviews and Team Culture

STRs can succeed only in an appropriate quality-focused culture. Members of a healthy culture regard reviews as constructive activities, not as opportunities to identify inferior performers or to find scapegoats for quality problems. STRs encourage authors to take pride in their craftsmanship because they know their peers will be scrutinizing their work.

The Influence of Culture

Key cultural factors that support or inhibit STRs are discussed. A commitment toward quality (in its many dimensions) and the recognition that team success depends on helping each other do the best job possible are essential. A literature example [Van Veenendaal] compares attitudes toward peer reviews among collaborating development teams in Singapore and The Netherlands. The human dynamics and behavioral issues of the work product author and other participants are described. Appropriate and inappropriate individual behaviors during reviews and the need for mutual respect and avoiding strained relationships are addressed. In a healthy culture, having a co-worker find a defect is regarded as a "good catch", not a personal failure. Holding a brief self-reflective team discussion at the end of each review to ask "How did we do?" and "How did we feel?" and "What prevented us from doing better?" is a culture-enhancing activity. A powerful stimulus for performing reviews could be a process improvement effort launched as a result of analyzing a recent painful failure. If influential resisters can be converted to believe in the value of STRs, they can persuade other team members to try them, too.

Why People Don't Do Reviews

Some of the reasons why software practitioners do not currently perform reviews are presented. These include the perception that they take too much time, previous unpleasant review experiences, the arrogant attitude that some people's work does not need reviewing, fear of management retribution or public ridicule if defects are discovered, and a reluctance to let others closely examine interim (or even completed) work products. If reviews aren't planned, spending time on them can seem to be a penalty for the reviewers and the producers, making the project slip by adding more useless work at the last minute.

Reviews and Management

Managers also need to be educated about STRs and their impact on the organization. Management attitudes toward STRs are explored, including political issues, the criminalization of bugs, and getting management committed to a review program. Several serious risks of using inspection results to evaluate team members are described.

Overcoming Resistance to Reviews

Several obstacles to doing inspections, sources of resistance, and ways to overcome them are presented. Suggestions about ways to instill inspections into different kind of organizational cultures (such as Constantine's four organizational paradigms, DeGrace

and Stahl's Roman and Greek cultural types, and Weinberg's six cultural patterns) are described.

Peer Review Sophistication Spectrum

A figure illustrates a scale of an organization's sophistication, or maturity, in its practice of peer reviews: no reviews are performed; random and unstructured reviews are performed; reviews are intended, scheduled, and structured; the review program is planned and managed and reviewers are trained; formal reviews are conducted as scheduled, they are viewed as critical contributors to project success, and the data collected are analyzed for process improvement.

Justifying Technical Reviews

Ample literature experience describes substantial returns on investment that various companies have actually achieved from inspections, ranging up to 10-to-1 [Ebenau, Gilb, Grady]. Even a modest ROI of 1.1 can make inspections worth doing. Some of these literature benefits and experiential examples are presented here. For example, one of the author's consulting clients spent an average of $200 to find and fix a defect by inspection, while it cost an average of $4,200 to correct a defect reported by a customer. A process for estimating the ROI in a specific organization based on the cost to find and fix defects by inspection versus by system testing or in the hands of the customer is described in Chapter 10.

Reviews provide additional, hard-to-quantify benefits. One is the knowledge that individuals gain from having their peers provide rapid feedback on their work so they can correct their behavior quickly, cross-training, and risk reduction by sharing information within the project team. Reviews help disseminate specific product, project, and technical knowledge among the team members, supplementing formal communication mechanisms. They also contribute to a collaborative mindset, with team members willing to share their knowledge, learn from others, and contribute to the quality of each system component and the final product. Those individuals who have experienced these intangible, but very real, benefits often clamor for reviews, because they know how much the reviews can help.

Planning for Reviews

Informal reviews are performed in an ad hoc fashion whenever the author solicits input from co-workers. Formal reviews should be planned as part of the project's life cycle, schedule, or work breakdown structure. Planning reviews is discussed, and the percentage of time a project should expect to spend on reviews and inspections is described. Reviews should be viewed not as milestones in the project plan, but as tasks, with a strong likelihood that time will have to be spent performing rework following each such quality control activity. Define exit criteria for key work products that include passing an appropriate technical review.

Guiding Principles for Reviews

Several guiding principles that contribute to successful reviews of any type are presented, including [Wiegers]:

- Check your egos at the door.
- Review the product, not the author.
- Limit review meetings to two hours.
- Find problems during reviews, but don't solve them.
- Use standard forms.
- Keep the review team small.
- Require advance preparation.

Chapter 3. The Technical Review Formality Spectrum (3000 words)

Chapter Objective: *To help the reader understand the variety of review methods available and their range of formality. Understand the value and limitations of different STR methods. How to select an appropriate STR technique for a particular situation.*

The Formality Spectrum

The general term "review" or "peer review" encompasses a variety of defect-detection techniques. The software professional's quality toolkit should include several different types of STRs, which fall along a spectrum of formality and comprehensiveness. A figure will illustrate this spectrum and show where several common STRs fit along the spectrum. Formal reviews are characterized by having a defined process, record-keeping, planning, metrics, specific participant roles, and trained participants. The IEEE 1028-1997 Standard for Software Reviews will be referenced to tie this formality spectrum to other established literature on reviews.

Inspection

The most formal type of STR is called an inspection. Inspection follows a well-defined, multi-stage process with specific roles assigned to individual participants. The published data indicates that inspections are more effective at finding defects than are less formal review techniques [Van Veenendaal]. Several types of inspection approaches have been developed, which are described in Chapter 4. Although they cost more than other review methods, the greater effectiveness can still render them cost-effective.

Team Review

Team reviews are essentially "inspection-lite," being well structured but somewhat less formal and less comprehensive than inspections. They follow many of the steps found in an inspection. Significant differences are that the reader role may be omitted, and the overview meeting and follow-up stages may be simplified or omitted.

Walkthrough

A walkthrough is a type of informal review in which the author of a work product describes it to a group of peers and solicits comments [Freedman]. This approach is often used during design reviews, when the objective is to assess whether the proposed design is sufficiently robust and appropriate to solve the problem. Walkthroughs differ substantially from inspections because the author takes the dominant role; no other specific roles or process are defined. Records usually are not kept. Walkthroughs can be an efficient way to review work products modified during maintenance.

Peer Deskcheck

In a peer deskcheck (pair reviewing, buddy check), only one person other than the author examines the work product. The author typically has no visibility into how the reviewer approached the task, so a wide variability in results is expected. This method is often suitable for low-risk work products or if you have colleagues who are especially skilled at finding defects on their own. A peer deskcheck is essentially the same as conducting only the preparation step of an inspection. A good way to begin participating in reviews is to find a colleague who you respect professionally and trust personally and exchange work products for peer deskchecks. This is good for mentoring, also. [Cusumano]

Passaround

The passaround is essentially a multi-person peer deskcheck. As an alternative to physically passing out multiple copies of the document to review, you can place an

electronic copy of the document in a shared folder and have reviewers contribute their comments in the form of document annotations, such as Microsoft Word or PDF comments.

Pair Programming

Pair programming is a component of Extreme Programming [Beck]. In pair programming, two developers work on the same product simultaneously at a single workstation [Williams]. This arrangement affords an opportunity for continuous, incremental, informal reviews of each other's work, which leads to superior work products by literally applying the old adage that "Two heads are better than one." This is a type of informal review because there is no preparation or documentation, a real-time review of code and tests. It doesn't provide the outside perspective of someone less personally attached to the code that a formal review brings.

Choosing a Review Approach

This section describes how to select an appropriate review method for use in a specific situation. Risk assessment (considering the likelihood and impact of having defects in the work product) can be used to decide whether a formal or informal review is most appropriate. Several factors are described that increase the risk associated with a given work product and hence make it more appropriate to use a formal review.

Chapter 4. The Inspection Process (5000 words)

Chapter Objective: *To describe the typical inspection process and roles, including different variations of the inspection method.*

Inspection Stages

The classical (Fagan) inspection process includes six discrete steps, not just a single meeting to discuss the work product [Ebenau]. The steps are: planning, overview, individual preparation, inspection meeting, rework, and follow-up. More advanced organizations with extensive inspection experience also perform defect causal analysis. A flow diagram illustrates this process. The details of each stage are described in detail in subsequent chapters. The Fagan inspection emphasizes finding defects during the inspection meeting, based on an understanding of the work product gained during the overview and preparation steps. Whether or not a meeting is really needed for an effective inspection is discussed [Votta].

Inspector Roles

The inspection team includes four specific participant roles: author (creator or maintainer of the work products being reviewed), moderator (plans and leads the inspection), reader (presents the material being inspected to the team during the meeting), and recorder (documents issues raised during the meeting). Any other participants are simply inspectors, with no special responsibilities. Inspection teams should generally be kept small, preferably from 3 to 7 participants. If the team is too large, it can be hard to cover the intended material at the expected rate, distracting side discussions can erupt, and the recorder has a harder time knowing what to capture. Keeping the inspection team small will reduce costs and improve efficiency, at the price of leaving some interested and possibly valuable participants out of the process. There is some literature evidence [Martin, Kosman] that holding multiple parallel inspections with very small teams (N-fold inspections) is more cost-effective than using a single large team. In a small team, the moderator could also serve as either recorder

or reader. A striking characteristic of inspection is that the author may *not* serve as reader, recorder, or moderator. The rationale behind this is explained.

Variations on the Inspection Theme

In addition to the classical inspection method described here, several other types of inspection have been proposed. The distinguishing characteristics of these alternative inspection approaches are briefly described here. A table of the terminology and inspector role names used in the different methods is included. In the author's opinion, the similarities between the methods outweigh the differences; diligently applying any of them is likely to give a development team substantial quality improvements.

Gilb Method

Tom Gilb's inspection method, described in his 1993 book with Dorothy Graham titled *Software Inspection*, is a rigorous, meticulous method that emphasizes measurement. Defects are found primarily during individual checking and are collected in a logging meeting. Although effective, the level of rigor and slow pace of covering the work product exceeds the tolerance for many organizations just starting out with STRs.

High-Impact Inspection

This inspection technique from Software Quality Engineering has slightly different roles and emphasizes the use of various analysis techniques to find defects. Few published results on the method's effectiveness are available.

Phased Inspections

This technique, from Knight and Myers, uses a series of coordinated, rigorous, partial inspections termed phases, which focus on determining whether a work product possess a specific desired property.

Bisant and Lyle Method

These authors describe a two-person inspection approach that does not use a moderator. It has been shown to improve programming productivity.

Chapter 5. Planning the Inspection (6500 words)

Chapter Objective: *Describe the participants and activities involved in planning an inspection, emphasizing the moderator role and characteristics.*

When to Hold Inspections

The appropriate timing of inspecting various kinds of work products is suggested. Inspections are often viewed as a final quality filter, a gate through which a work product must pass before being approved or baselined. This is appropriate, but inspections should not be limited to being a final stamp of approval. Inspections can be combined with earlier or incremental, informal reviews to provide an efficient and cost-effective quality approach.

Assembling the Cast

When the author has a work product ready for inspection, the first step is to choose a moderator. Several ways to select a moderator are suggested [Ebenau]. The author should describe his objectives for the inspection to the moderator. The author and moderator then choose other suitable participants and assign two of them to the roles of recorder and reader. Characteristics of effective reviewers are summarized.

The Inspection Moderator

The moderator plays a vital role in having a successful inspection. The moderator's role, characteristics, skills, and behaviors are described. Moderators should receive specific training in how to perform this challenging task. The risks of having too few qualified moderators available are pointed out.

Inspector Perspectives

The inspection participants should represent four key perspectives, illustrated in a figure: peers of the author; authors of any predecessor work products or specifications for the product being inspected; (e.g., designers for a code inspection); anyone whose work must interface with the product being inspected; and anyone who will have to do downstream work based on the product being inspected (e.g., developers or testers for a requirements specification inspection). Inspectors having these different perspectives and interests will detect different kinds of issues and defects.

What About Managers and Observers?

The common wisdom is that managers do not belong in inspections. However, I think this is culturally dependent on the degree of mutual respect among the manager and technical team members. The author's first-level manager could participate if this is acceptable to the author and if the manager has the technical knowledge to identify defects. Too, managers need to have their own work products reviewed, so simply excluding managers from inspections isn't sensible. Observers should generally be restricted, although inviting interested people to the overview meeting is a way to help them be involved and informed about the work products. Having new moderators observe an experienced moderator in action can be a useful learning experience.

Selecting the Material

The author and moderator can determine whether to examine the entire work product or just selected portions. Sampling is appropriate for large work products that cannot be examined in their entirety, for products that have sections of different complexity or technical risk, when time is limited, or to obtain quality data that will indicate whether the entire product needs to be examined. Suggestions are presented about the most critical parts of the work product to inspect [Freedman].

Inspection Entry Criteria

Entry criteria define the conditions that must be satisfied prior to inspecting a specific type of work product. If inspection proceeds without satisfying the entry criteria, the likelihood of holding an optimally effective inspection is reduced [Gilb]. Reference is made to the Appendix, where entry criteria are proposed for a variety of software work products.

The Inspection Package

During planning, the author and moderator assemble a package of materials for each inspector. This package includes the work product being examined, any associated predecessor specifications, test documentation that relates to the product being inspected, any necessary forms or checklists, and any other pertinent documents that the inspectors need to do a thorough job. Some inspections examine the work product and corresponding test documents together, thereby potentially finding defects in both.

Inspection Rates

Part of planning is to determine how much material can likely be inspected in the time available. The examination and inspection rates for different kinds of work products have

been quantified in several publications. An inverse relationship is always found between the defect density (defects found per unit of material, such as per thousand lines of code) and the inspection rate. Data on the optimum preparation and inspection rates, and an evaluation of the risk of having undiscovered defects remain in the product, help the moderator and author decide how quickly to evaluate the material for each inspection.

Chapter 6. Using the Eagle-Eye (3500 words)

Chapter Objective: *Describe the inspection overview meeting and activities involved in individual preparation, including analysis techniques for finding errors.*

The Overview

The overview step brings all inspection participants up to speed on the purpose, context, and history of the work product, including any assumptions the author has made. This is often conducted as an optional overview meeting led by the author. The primary purpose of this step is education, not defect-detection. If you have many interested people who want to learn about the product but you don't want them to participate in the inspection meeting, you could invite them to the overview meeting instead. If the rest of the team is already sufficiently knowledgeable about the work product, you can skip the overview meeting. A meeting might not even be necessary: the author might be able to provide sufficient background information through a cover letter on the inspection package.

Individual Preparation

During individual preparation, each inspector (including the author) examines the work product to understand it and to find possible defects. Up to 75 percent of all the defects found by inspection can be found during preparation [Humphrey]. Suggestions are provided for how to examine different kinds of work products during individual preparation. Sometimes preparation is done as a "study hall" session immediately following the overview meeting. The question of whether inspectors should ask the author questions during preparation is addressed. Guidelines are presented for determining the appropriate preparation time, based on statistical data [Ebenau, Weller]. An alternative approach to preparation is described, in which the issues found during examination are provided to the author prior to the inspection meeting so the author can address them specifically during the meeting and identify which issues were accepted and corrected as defects, and which were not. Minor defects found during preparation, such as typographical errors, can be noted on a Minor Fault Summary Sheet that's given to the author, so time need not be spent on these small issues during the inspection meeting.

Inspection Checklists

Checklists that identify the kinds of mistakes often made in different work products are an important part of an organization's inspection infrastructure. Checklists help the inspectors focus their attention during individual preparation on the most likely sources of errors. Checklists also help team members create better products by knowing in advance what kind of problems to watch out for. The Appendix contains sample checklists for several types of work products. Readers are encouraged to acquire electronic copies of these checklists from the author's website and tailor the checklists to meet their specific project needs.

Analysis Techniques

Several analysis methods appropriate for various life cycle stages and work products are described [Gelperin]. Inspectors can apply these analysis methods during their individual preparation. The team should concentrate on finding errors that could cause failures that

are hard to observe. Risk assessment thinking can be used to identify possible failure modes and to select suitable inspection analysis strategies.

Assigning Preparation Responsibilities

It's often a good idea to parcel out specific preparation activities to different inspectors, such as checking cross-references, checking interfaces, consistency checking on variable names or terminology, checking the work product against pertinent standards, and tracing work product elements back to requirements. Selecting suitable analysis strategies, rather than leaving the preparation mode up to each individual, helps reduce redundancy and focuses the participants' energy on the specific areas of concern.

Chapter 7. Putting Your Heads Together (6000 words)

Chapter Objective: *Describe the inspection meeting, including roles and activities and the moderator's activities.*

Launching the Inspection Meeting

The inspection meeting is the heart of each inspection. The moderator controls the meeting, beginning by introducing the participants, stating the inspection goals, and collecting each inspector's preparation time. If the moderator judges that insufficient preparation has been done, the meeting should be rescheduled for a later time. Several ways to judge whether preparation is adequate will be suggested.

Conducting the Meeting

Reading the Work Product

The reader has a lead role during the inspection meeting. The reader paraphrases the work product one section at a time, essentially describing it at a higher level of abstraction than the verbatim material. The merits and shortcomings of using the reader are described, as are some techniques for reading various types of work products, based in part on recent research literature [Basili].

Raising Issues

Inspection participants can point out issues and defects in several ways. Most commonly, the reader pauses briefly after presenting each section of material to permit inspectors to offer observations. Alternatively, the team can use a round-robin approach, asking each inspector in turn for their issues. Sample dialogs are included to illustrate the flow of the inspection meeting. Ways that participants can phrase issues and defects inoffensively are suggested, to keep the inspection focused on a constructive mindset and reduce the author's defensiveness.

The Recorder's Role

The inspection recorder has the challenging job of capturing the essence of the issues discussed in the meeting in a concise and inoffensive way. The recorder uses a standard Peer Review Issues List form. The recorder should echo what he or she has written to the team aloud to verify correctness. The recorder also performs an initial defect classification; classification schemes are discussed briefly.

Moderator Roles and Tools

The moderator is the process owner for the inspection meeting. He or she is responsible for keeping the meeting on track, correcting any inappropriate personal behaviors, and limiting

discussions that threaten to sidetrack the meeting into problem-solving or decision-making. Occasionally, a situation might arise in which the focus of the meeting should change from defect-detection to problem-solving, as when a fundamental flaw that undermines the whole premise of the work product is detected. The moderator might need a device to get the attention of the participants from time to time, and the other inspectors need to respect the moderator's authority to keep the inspection meeting focused on its objectives. Other work aids and tools, such as checklists, for the moderator are also described.

Product Appraisal

At the end of the inspection meeting, the participants decide on an appraisal of the work product: accepted as is, accepted with minor revision, major revision and re-inspection needed, or inspection not completed. The team should decide in advance how they will select this appraisal. One possibility is that, if the team members don't agree, the most conservative appraisal expressed by any of the inspectors is assigned to the product [Gelperin].

Inspection Meeting Problems

Several things that can go wrong during the inspection meeting are described, along with suggestions for how to deal with them. These problems include lack of participation by attendees, personal attacks on the author, a moderator who fails to control the meeting, being distracted by problem solving, the recorder not keeping up, and reader going too fast or too slowly.

Improving the Inspection Process

Because process improvement is a never-ending activity, it's a good idea for the moderator to close the inspection meeting collecting comments from the participants on this inspection experience and how it could be improved. A checklist of such closure questions is presented.

Chapter 8. Bringing Closure (2000 words)

Chapter Objective: *Describe the inspection activities that follow the meeting, including rework to correct defects, follow-up to verify rework and issue closure, and, for more mature organizations, defect causal analysis. The exit criteria for declaring an inspection complete are described.*

Rework

Nearly every inspection will identify some defects that should be corrected, as well as improvement suggestions that the author may or may not choose to implement. The author needs to resolve every issue raised, although he or she might decide not to correct every defect found. The author is ultimately responsible for the work product's quality, but possible reasons why it might be okay not to correct a defect are presented.

Follow-Up

The follow-up step brings the inspection to closure by providing another pair of eyes to verify that all issues from the inspection meeting were resolved appropriately and that any changes made in the work product were made correctly. The type of follow-up activity is determined by the work product appraisal that the inspection team established at the close of the meeting. In some cases follow-up might be waived, in others the moderator or someone else will review the changes made, and in still other cases a partial or complete re-inspection might be needed. A group smaller than the original inspection team might perform the re-review, or the entire inspection team could examine the reworked product before deciding whether another full inspection is needed.

Causal Analysis

Studying the patterns of defects found in different kinds of work products and understanding their root causes provides a powerful tool for process improvement. The example of the Space Shuttle on-board software is cited [Paulk], in which all defects found had their causes identified, the development process that led to the defect corrected, and the quality control processes improved to detect similar defects earlier. A related activity is to analyze errors that leaked through the project's various quality filters to see if they could have been found earlier by inspections or testing. These insights can lead to improving the inspection process and the checklists used. The type and frequency distribution of errors detected by inspection can lead to suggestions for changing an organization's processes to reduce the occurrence of such errors. The inspection data can also suggest ways to improve the inspection process itself to improve its error-detection efficiency and effectiveness.

Inspection Exit Criteria

An inspection is completed when the exit criteria for that inspection (not just the inspection meeting) are satisfied. Typical exit criteria are presented, including each issue being resolved, any uncorrected defects being recorded in the project's defect-tracking system, and the modified work product being checked into the project's configuration management system.

Chapter 9. Installing a Review Program (4500 words)

Chapter Objective: *Describe the organizational characteristics and activities that are needed to implement a successful technical review program.*

Preparing the Organization

Several components that must be present for an organization to successfully adopt an STR program are described. Recommendations are provided for laying the foundation for success and for implementing the program in a way that maximizes the beneficial impact on quality but minimizes the adverse impact on tight project schedules. Suggestions are provided for piloting a new review process before rolling it out to the entire organization.

Process Assets

Several key process assets that are needed for an organization to implement effective reviews are described. These include a peer review process description, necessary forms, checklists of defects found in different kinds of work products, and data collection and analysis tools. A sample peer review process description that contains both informal review and inspection procedures is included in the appendix and on the author's website.

The Reviews Process Owner

It's a good idea for an organization to identify a management-level process owner for its STR program. Some responsibilities of such a process owner are suggested.

The Review Coordinator

An organization that is serious about its STR program will identify an individual to serve as their review coordinator. The coordinator collects and analyzes data from multiple reviews, offers suggestions for improving both the STR processes and the organization's development processes based on STR data analysis, coaches team members on practicing STRs, and works with the process owner to make the STR program as effective as possible. The review

coordinator can answer the ultimate question about a review program's effectiveness: are you finding bugs?

Training Review Participants

Every potential participant in STRs should receive basic training in the concepts and practices of formal and informal STRs, as well as in the specific STR processes used in their organization. People who will serve as moderators should receive some additional training, emphasizing practice sessions. The major topics to cover in such training classes are suggested. Training isn't enough to guarantee a successful review program, so suggestions are provided about how to follow up after training to reduce the concepts to practice and to judge how well your process is working.

Piloting the Review Process

New processes often look good on paper but need some adjustment in practice. Plan to pilot your review process (perhaps on the process description itself!) before inflicting it on the organization. A pilot could involve a few early adopters who are willing to follow the new process very carefully to see how well it works in operation. The process owner is responsible for seeing that the process is appropriately modified based on the pilot experience.

Chapter 10. Keeping Review Records (4000 words)

Chapter Objective: *Describe three levels of record-keeping for STRs: using standard forms, collecting information from multiple reviews in a database, and analyzing process trends from a series of reviews. Help the reader understand the value of collecting and using such data.*

Review Forms

Several standard STR forms are illustrated. These will also be made available in electronic form through the author's website (in fact, most are already there). The key forms are: review meeting notice, formal review summary report, review issues list, and minor fault summary sheet. The scope of visibility of inspection data is described: the author owns the issues list, while managers, process engineers, and all team members can see summaries.

Technical Review Database

An organization can collect the results of multiple reviews to develop process metrics on preparation and inspection rates, defect densities in source code and other work products, and the time (and hence cost) invested in STRs. A set of simple spreadsheets that the reader can use to collect such data will be made available on the author's website.

Statistical Process Control

When a consistent inspection process has become well-established in an organization, it becomes possible to apply statistical process control (SPC) to quantify key inspection parameters. The learnings from several published articles on applying SPC to inspection data [Ebenau, Weller, Williams] are summarized to illustrate the most sophisticated current capability of software process management.

Measuring the Impact of Reviews

Participants often can judge how valuable reviews are for themselves, but often some more quantitative benefit analysis is needed. One outcome of an effective STR program should be

that fewer defects are found during later stage testing. However, there's a cultural issue here, too. If the testers find fewer bugs, someone might inappropriately conclude that testing is less effective than it used to be. A technique for estimating the quantitative benefits of reviews (particularly inspections) in an organization is provided. This technique compares the costs invested in an STR program (including training, developing and maintaining the infrastructure, performing reviews, analyzing data, and process management) with the estimated cost savings. Savings are estimated from the counts of bugs found that might otherwise not have been found until a later life cycle phase and the cost amplification of finding the defects later. This analysis helps organizations estimate their savings from inspections and determine whether it's worth performing them.

Chapter 11. Making Reviews Work for You (5000 words)

Chapter Objective: *Summarize key success factors for implementing and sustaining STRs in an organization. Present indicators that warn that a review program is struggling.*

Critical Success Factors for Reviews

- Adopt this attitude: "We prefer to have a peer, rather than a customer, find a defect" [Wiegers].
- Allocate time and resources in the project plan for reviews and rework.
- Conduct training for all reviewers.
- Review your early reviews, to improve their effectiveness.
- Review early and often, formally and informally.
- Become a local champion for reviews; coach and encourage others.

Symptoms of a Struggling Review Process

Several indications that an organization's review program is floundering are described, along with suggestions for dealing with each kind of problem. This section can be used as a troubleshooting guide for readers who want to improve their current review activities.

Review Traps to Avoid

Seven common traps that undermine the success of an STR program are described here, along with suggestions (or references to other parts of the book) for avoiding the traps or extricating yourself from them [Wiegers]. Most of these points were brought up earlier, but they are summarized and reinforced here. The traps are:

- participants don't understand the review process
- the review process isn't followed
- participants critique the producer, not the product
- reviews are not planned
- review meetings drift into problem solving
- reviewers are not prepared
- the wrong people participate
- reviewers focus on style, not substance

Habits of Effective Reviewers

One sign of an effective review process is that people complain about the quality of work products and missing documents. As you become accustomed to higher quality products, you will become less tolerant of low quality and incomplete products. This closing section summarizes several characteristics and habits of effective STR participants [Allott].

Chapter 12. Special Review Challenges (2000 words)

Chapter Objective: *Discuss current thinking about inspections and technical reviews that might be at odds with the traditional understanding. Address special inspection situations, such as non-procedural code, large work products, and distributed development teams.*

Certain kinds of work products or team situations pose special challenges to the organization's STR efforts. Some of these challenges and possible ways to address them are the topic of this section.

Large Work Products

Many organizations are creating very large work products with thousands of requirements and hundreds of thousands of lines of code. It might not be realistic to inspect all of these work products, although the organization should consider carefully whether the benefits might not outweigh the costs. You can use a risk analysis to select those portions of the product for which undetected errors could cause the most problems in the future. Sampling portions of large work products can give an indication of the overall expected quality and help you judge whether it's beneficial to look carefully at more than just a few samples. When reviewing large documents, consider asking different reviewers to start their examination at different pages, rather than having only the first few pages studied closely with fresh eyes. Another alternative is to use multiple parallel reviews of different parts of the work product by smaller groups.

Geographic Separation

As teams collaborating across long distances are doing more and more software development, the traditional face-to-face inspection method must be modified. Some Internet-based collaboration tools are available to facilitate remote technical reviews. The author has participated in both tele- and videoconference inspections, which place special burdens on the participants and on the moderator, such as the absence of body language and facial expressions and difficulty getting the participants' attention.

Non-Procedural and Generated Code

There is little point in manually reviewing non-procedural code and code generated by 4GLs, GUI builders, visual development tools, and the like. The reviews should focus instead on the designs or other inputs that these tools used to generate the code, making the assumption that the code generator function proceeds correctly under all circumstances.

Too Many Participants

Suggestions are presented for handling situations in which a large number of participants wish to be involved in an inspection or many different project stakeholders are affected by the work product being inspected.

No Qualified Reviewers Available

If your group doesn't have anyone with the technical knowledge to review some team member's work, you have a risk of that individual's specialized knowledge departing in these days of high software staff turnover. You might begin holding reviews as cross-training to impart essential knowledge to other team members, who will gradually become able to find possible defects. Also consider working with suitably skilled colleagues in other departments or companies, possibly using e-mail to initiate remote reviews if necessary.

Appendix A. Peer Reviews and the CMM (2000 words)

Chapter Objective: *Describe what the Capability Maturity Model for Software says about the Peer Reviews Level 3 key process area and other types of reviews. Describe the practical implications of following the Peer Reviews KPA.*

Reviews and the CMM

The expectation of holding reviews appears throughout the CMM, in all key process areas (KPAs) [Paulk]. These include status reviews, management reviews, and quality reviews. A formal peer review program is required for an organization to achieve CMM level. However, all organizations can benefit from practicing STRs, whether or not they are using the CMM to guide their software process improvement activities. This chapter describes the key practices of the Peer Reviews KPA. The rationale behind why peer reviews are addressed at Level 3 of the CMM is summarized.

Goals of the Peer Reviews KPA

The CMM states two to four goals for each KPA. The two goals for Peer Reviews are:

1. Peer review activities are planned.
2. Defects in the software work products are identified and removed.

Peer Reviews Activities Performed

The CMM identifies several activities that an organization should perform to implement peer reviews. These activities include: peer reviews are planned, and the plans are documented; peer reviews are performed according to a documented procedure; data on the conduct and results of the peer reviews are recorded. These activities make good sense whether or not your organization is striving to achieve CMM Level 3.

Institutionalizing Peer Reviews

On the surface, performing these activities will achieve the KPA goals, but it's also important to lay a foundation for sustained performance of their software technical reviews. To this end, the CMM recommends a set of practices to help "institutionalize" peer reviews in an organization.

Commitment to Perform

An organization demonstrates a serious commitment to perform peer reviews when senior management issues a policy that states their expectations about reviews. Members of the organization should be held accountable for conforming to the policy.

Ability to Perform

It's not realistic to expect an organization's peer review program to succeed unless its members are given the ability to perform what is expected of them. Enabling this performance means providing adequate time, resources, tools, and training.

Measurement and Analysis

An organization should collect and analyze data on the way it performs peer reviews. Some suggestions for appropriate data items and how to collect and analyze them are given in Chapter 10.

Verifying Implementation

Even with the best of intentions, it's a good idea to have an external, objective observer verify that the members of the organization are in fact performing peer reviews as intended, that they have sufficient and appropriate procedures in place, and that the reviews are effective at finding defects. This process verification is the province of a software quality assurance group.

Appendix B (6000 words) [on web]

Checklists of common errors in different kinds of work products

Entry and exit criteria for different work products

Peer review process description

