



Why Should Your Team Conduct Peer Reviews?

By Linda Westfall

lwestfall@westfallteam.com

What is a Peer Review?

The IEEE/ISO/IEC Systems and Software Engineering Vocabulary defines a review as “a process or meeting during which a work product, or set of work products, is presented to project personnel, managers, users, customers, or other interested parties for comment or approval.” [IEEE/ISO/IEC 2017]

A peer review is a special type of technical review where one or more of the Author’s peers evaluate a work product to identify defects, obtain a confidence level that the product meets its requirements, and/or identify opportunities to improve that work product. The Author of a work product is the person that either originally produced that work product or the person who is currently responsible for maintaining that work product.

Peer Review Objectives

One of the primary objectives of peer reviews is to identify and remove defects in software work products as early in the software life cycle as possible. Sometimes it can be very difficult for the Author to find defects in their own work product. Most software practitioners have experienced situations where they hunted and hunted for that elusive defect and just couldn’t find it. When they ask someone else to help, the other person takes a quick look at the work product and spots the defect almost instantly. That’s the power of peer reviews.

Another objective of peer reviews is to provide confidence that the work product meets its requirements (verification) and the customers’ needs (validation). Peer reviews can analyze the work product to ensure that all of the functional requirements, quality attributes and other non-functional requirements have been adequately implemented in the design and code or are adequately being evaluated by the tests.

Peer reviews can be used to check the work product for compliance to standards. For example, the design can be reviewed to ensure that it matches modeling standards and notations, or the code can be reviewed to ensure that it complies with coding standards and naming conventions.

Peer reviews can also be used to identify areas for improvement (this does not mean “style” issues, if it is a matter of style, the Author wins). However, peer reviewers can

identify areas that make the software more efficient., maintainable, testable, and so on. For example, when reviewing a piece of source code, a Reviewer might identify a more efficient sorting routine, a method of removing redundant code or even identify areas where existing code can be reused. During a peer review, a tester might identify issues with the testability of a requirement or section of the design. Reviewers can also identify maintainability issues. For example, in a code review, inadequate comments, hard-coded variable values, or confusing code indentation might be identified as areas for improvement.

Benefits of Peer Reviews

The benefits of peer reviews, especially formal inspections are well-documented in the industry. For example, more defects are typically found using peer reviews than other verification and validation (V&V) methods. Capers Jones reports, "Within a single testing stage, you are unlikely to remove more than 35% of the errors in the tested work product. In contrast, design and code inspections typically remove between 50% and 70% of the defects present." Well-run inspections with highly experienced Inspectors can obtain 90% defect removal effectiveness. [Wiegers-02] "Inspections can be expected to reduce defects found in field use by one or two orders of magnitude." [Gilb-93]

It typically takes much less time per defect to identify defects during peer reviews than it does using testing techniques. For example, Kaplan reports that at IBM's Santa Teresa laboratory, it took an average of 3.5 labor hours to find a major defect using code inspection while it took 15 to 25 hours to find a major defect during testing. [Wiegers-02]

It also typically takes much less time to fix them because the defect is identified directly in the work product, which eliminates the need for time-consuming debugging activities.

Peer reviews can also be used early in the life cycle on work products such as requirements and design specifications to eliminate defects before those defects propagate into other work products and become more expensive to correct.

Peer reviews can also be beneficial because they help provide opportunities for cross-training. Less experienced practitioners can benefit from seeing what a high-quality work product looks such as when they help peer review the work of more experienced practitioners. More experienced practitioners can provide engineering analysis and improvement suggestions that help transition knowledge when they peer review the work of less experienced practitioners. Peer reviews also help spread product, project, and technical knowledge around the organization. For example, after a peer review, more than one practitioner is familiar with the reviewed work product and can potentially support it if its author is unavailable. Peer reviews of requirements and design documents aid in communications and helps promote a common understanding of those requirements that is beneficial in future development activities. For example, peer reviews can help identify and clarify assumptions or ambiguities in the work products being reviewed.

Peer reviews can help establish shared workmanship standards and expectations. They can build a synergistic mindset as the work products transition from individual to team ownership with the peer review.

Finally, peer reviews provide data that aid the team in assessing the quality and reliability of the work products. Peer review data can also be used to drive future defect prevention and process improvement efforts. For example, through the root cause analysis of defects identified during peer reviews.

What to Peer Review

Every work product that is created during software development can be peer reviewed. However, not every work product should be. Before a peer review is held, practitioners should ask the question, “Will it cost more to perform this peer review than the benefit of holding it is worth?” Peer reviews, like any other process activity, should always be value-added or they should not be held. I typically recommend that every work product that is delivered to a customer or end-user be considered as a candidate for peer reviews. Examples include responses to requests for proposals, contracts, user’s manuals, requirement specifications and, of course, the software and its subcomponents.

In addition, any work product that is input to or has a major influence in the creation of these deliverables should also be peer reviewed. For example, a low-level design, interface specification, test cases/procedure, or automation scripts may never get directly delivered but defects in those documents can have a major impact on the quality of the delivered software.

Other internally used work products can be peer reviewed on a value-added basis. For example, some teams choose to peer review their project planning documents (project, quality, configuration management, verification and validation, risk management, safety plans, and so on) and other teams do not. Process documents and work instructions are another candidate for value-added peer reviews.

So, what doesn’t get peer reviewed? Actually, many work products are created in the process of developing software that may not be candidates for peer reviews. For example, it is typically not value-added to peer review quality records such as meeting minutes, status reports, and defect logs.

References:

- Gilb-93 Tom Gilb, *Software Inspections*, Addison-Wesley, Wokingham, England, 1993.
- IEEE/ISO/
IEC 2017 *IEEE/ISO/IEC 24765 Systems and Software Engineering—Vocabulary*.
Geneva, Switzerland: ISO. New York, NY: The Institute of Electrical and
Electronic Engineers.
- Wieger-02 Karl Wieger, *Peer Reviews in Software*, Addison-Wesley, Boston, 2002.