# Rise-Based Peer Reviews

Linda Westfall
The Westfall Team
lwestfall@westfallteam.com
www.westfallteam.com

## Abstract

While the benefits of formal inspections are well documented, in reality, many projects don't have the resources to formally peer review every work product at that level of rigor. By analyzing the probability that work product contains defects that will escape from development into the user environment and the potential impact of those defects, developers can make strategic, risk-based trade-offs between efficiency and effectiveness in their peer review processes. These decisions include the type(s) of peer review, the level of formality, the number of participants and peer review sufficiency needed for each work product.

## Introduction

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-610] 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, to obtain a confidence level that the product meets its requirements and/or to 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.

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 and the customers' needs. Peer reviews can analyze the work product to ensure that all of the functional requirements and quality attributes 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 peer 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. For example, when reviewing a piece of source code, a Reviewer might identify a more efficient sorting routine or 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 any of the other defect detection 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 specification 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 that 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 aids in communications and helps promote a common understanding that is beneficial in future development activities. For example, these 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 mind-set 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.

**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 candidates 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 major influence in the creation of these deliverables should also be peer reviewed. For example, a low-level design, interface specification or test cases may never get directly delivered but defects in those documents can have a major impact on the quality of the delivered software.

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 most quality records such as meeting minutes, status reports, and defect logs.

**Informal vs. Formal Peer Reviews**

Peer reviews can vary greatly in their level of formality. At the most informal end of the peer review spectrum, a software practitioner can ask a colleague to, "Please take a look at this for me." These types of informal peer reviews are performed all of the time. It is just good practice to get a second pair of eyes on a work product when the practitioner is having problems or needs a second opinion. As illustrated in Figure 1, these informal reviews are done adhoc with no formal process, no preparation, no quality records or metrics. Defects are usually reported either verbally or as redlined mark-ups on a draft copy of the work product. Any rework that results from these informal peer reviews is up to the author's discretion.

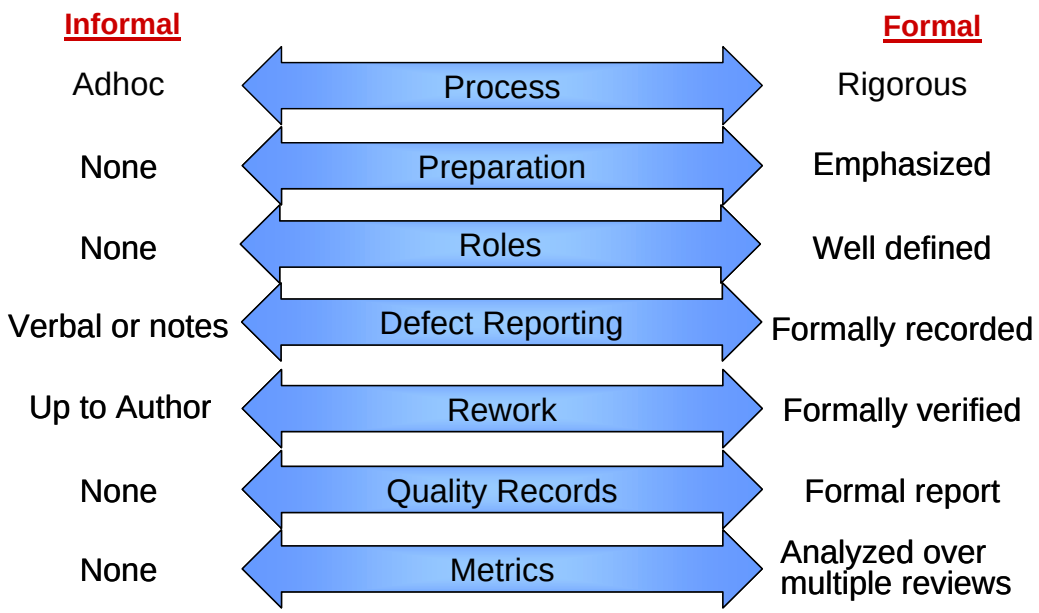



**Figure 1: Informal vs. Formal Peer Reviews**

On the opposite side of the spectrum is the formal peer review. In formal peer reviews, a rigorous process is documented, followed and continuously improved with feedback from peer reviews as they are being conducted. Preparation before the peer review meeting is emphasized. Peer review participants have well defined roles and responsibilities to fulfill during the review. Defects are formally recorded and that list of defects and a formal peer review report become quality records for the review. The author is responsible for the rework required to correct the reporting defects and that rework is formally verified by either re-reviewing the work product or through checking done by another member of the peer review team (for example, the inspection moderator). Metrics are collected and used as part of the peer review process. Metrics are also used to analyze multiple reviews over time as a mechanism for process improvement and defect prevention.

**Types of Peer Reviews**

There are many different types of peer reviews called by many different names in the software industry. Peer reviews go by names such as inspections, team reviews, technical reviews, walk-throughs, pair reviews, pass-arounds, adhoc reviews, desk checks and others. However, I have found that most of these can be classified into one of three major peer review types:

- **Desk Check**: A desk check is the process where one or more peers of a work product's Author reviews that work product individually. Desk checking can be done to detect defects in the work product and/or to provide engineering analysis. The formality used during the desk checking process can vary. Desk checking can be the most informal of the peer review processes or more formal peer review techniques can be applied. Desk checking can be a complete peer review process in and of itself, or it can be used as part of the preparation step for a walk-through or inspection.
- **Walk-Throughs**: A walk-through is the process where one or more peers of a work product's Author meet with that Author to review that work product as a team. A walk-through can be done to detect defects in the work product and/or to perform engineering analysis. The formality used during the walk-through process can also vary. An example of a very informal walk-through might be an Author holding an impromptu "white board" walk-through of an algorithm or other design element. In an informal walk-through there may be little or no preparation. In a more formal walkthrough, preparation is done prior to the team meeting typically through the use of desk checking. Typically preparation is left to the discretion of the individual Reviewer and may range from little or no

preparation to an in-depth study of the work product under review. During the walk-through meeting, the Author presents the work product one section at a time and explains each section to the reviewers. The Reviewers ask questions, make suggestions (engineering analysis) or report defects found. The Recorder keeps a record of the discussion and any suggestions or defects identified. After the walk-through meeting, the Recorder produces the minutes from the meeting and the Author makes any required changes to the work product to incorporate suggestions and to correct defects.

- **Inspections**: An inspection is a very formal method of peer review where a team of peers, including the Author, performs detailed preparation and then meets to examine a work product. The work product is typically inspected when the Author thinks it is complete and ready for transition to the next phase or activity. The focus of an inspection is only on defect identification. Individual preparation using checklists and assigned roles is emphasized. Metrics are collected and used to determine entry criteria in the inspection meeting as well as for input into product/process improvement efforts.

Figure 2 illustrates that while inspections are always very formal peer reviews, the level of formality in desk checking and walk-throughs varies greatly depending on the needs of the project, the timing of the reviews and the participants involved.

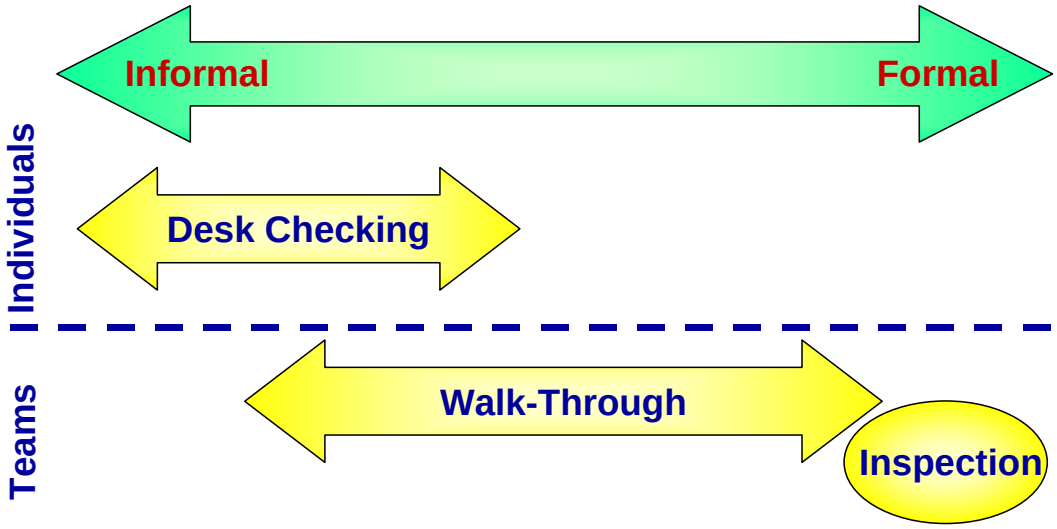



**Figure 2: Types of Peer Reviews**

The type of peer review that should be chosen depends on several factors. First, inspections are focused purely on defect detection. If the Author is looking for engineering analysis and improvement suggestions (for example, reducing unnecessary complexity, suggesting alternative approaches, identifying poor methods or areas which can be made more robust), a desk check or walk-through should be used. The maturity of the work product being reviewed should also be considered when selecting the peer review type. Desk check or walk-throughs can be performed very early in the life of the work product being reviewed. For example, as soon as the code has a clean compile or a document has been spell checked. In fact, white-board walk-throughs can be used just to bounce around very early concepts before there even is a work product. However, inspections are performed once the author thinks the work product is done and ready to transition into the next phase or activity in development. Staff availability and location can also be a factor. If the peer review team is geographically dispersed, it can be much easier to perform desk-checks than walk-throughs or inspections. However, the use of modern technology including web-based meetings and video conferencing makes long distance peer review meetings much more feasible. Economic factors such as cost, schedule and effort should also be considered. Team reviews tend to cost more and take longer than individuals reviewing separately. More formal peer reviews also tend to cost more and take longer. However, the trade-off is the effectiveness of the reviews. Team peer reviews take advantage of team synergy to find more defects and more formal

reviews also typically are more thorough and therefore more effective at identifying defects. The final factor to consider when choosing which type of peer review to hold is risk.

**Risk-Based Peer Reviews**

Risk-based peer reviews focus on the identification of software work products with the highest risk exposure. In risk-based peer reviews, risk probability is the estimated likelihood that a yet undiscovered, important defect will exist in the work product after the completion of the peer review. Multiple factors or probability indicators may contribute to a work product having a higher or lower risk probability. These probability indicators may vary from project to project or from environment to environment. Therefore, each organization or project should determine and maintain a list of probability indicators to consider when assigning risk probabilities to its software work product. Figure 3 illustrates examples of probability indicators, including:

- Requirements Stability: The more churn there has been in the requirements allocated to the software item, the more likely it is that there are defects in that item
- Past Quality: Software items that have had a history of defects in the past are more likely to have additional defects
- Size and Complexity: Larger and/or more complex software items are more likely to have defects than smaller and/or more simple software items
- Quality Attributes: The higher the constraints on the quality attribute (for example, reliability, performance, safety, security, maintainability), the more likely it is that there are related defects in that software item
- Operational Profile: The more used an item in the software is the more likely it is that the users will encounter any defects that might exist in that part of the software
- Author Knowledge and Skill: Novice software developers with less knowledge and skill tend to make more mistakes than experienced developers resulting in more defects in their work products

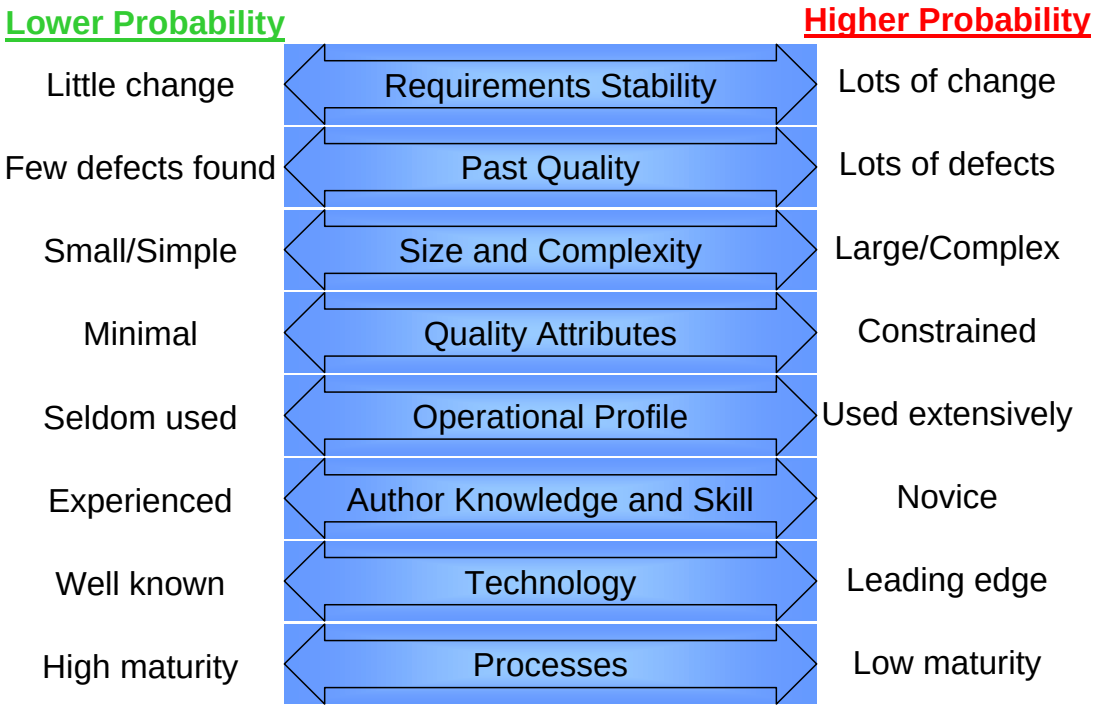



**Figure 3: Probability Indicators - Examples**

- Technology: If the developer is very familiar with the programming language, tool set and business domain they are less likely to make mistakes than if they are working with new or leading edge technology
- Processes: Higher maturity processes are more likely to help prevent defects from getting into the work products

Risk-based peer reviews also analyze risk impact, which is the estimated cost of the result or consequence if one or more undiscovered defects escape detection during the peer review. Again multiple and varying factors or impact indicators may contribute to a work product having a higher or lower risk impact. Each organization or project should determine and maintain a list of impact indicators to consider when assigning risk impacts to its work products. Examples of impact indicators include:

- Schedule and effort impacts
- Development and testing costs
- Internal and external failure costs
- Corrective action costs
- High maintenance costs
- Customer dissatisfaction or negative publicity
- Lost market opportunities
- Litigation, warranty costs or penalties
- Non-compliance with regulatory requirements

**Risk-Based Choices**

Once the risk probability and risk impact are analyzed, the risk exposure is determined and used to make risk based decisions about conducting the peer review. These decisions include:

- **Type of Peer Review and Level of Formality:** As illustrated in Figure 4, if there is both a low probability and a low impact (low risk exposure), then informal desk checking may be appropriate, as illustrated in Figure 4. As the probability and impact increase, the type of appropriate peer reviews moves through more formal desk checking to informal walk-throughs to more formal walk-throughs to formal inspections. For work products with a very high very high risk, having multiple peer reviews may be appropriate. For example, a product may be desk checked or have a walk-through early in development, and then be inspected late in its development just before it is released.

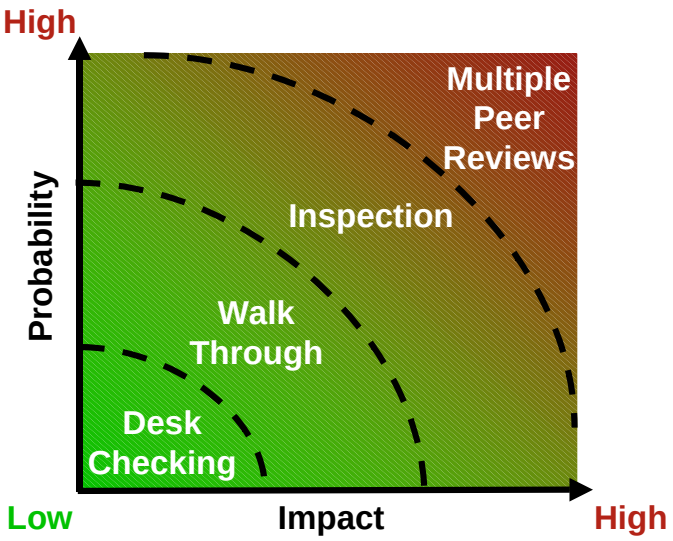
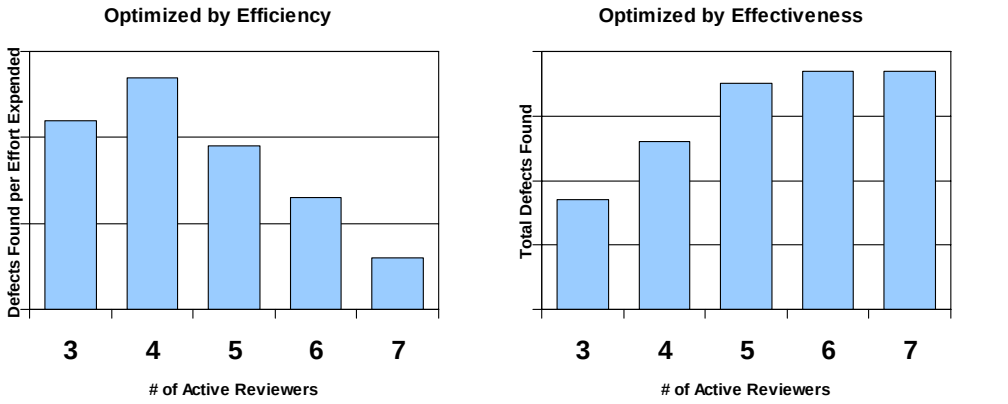



**Figure 4: Risk-Based Selection of Peer Review Types**

- **Number of participants**: The number of people performing the peer review may also be varied based on risk. For very low risk work products, having a single individual perform a desk check may be appropriate. For slightly higher risk work products, it may be appropriate to have multiple people perform the desk check. For products worthy of an investment in an inspection, less risky work products may be assigned a smaller inspection team of 2-4 people and higher risk products may be assigned an inspection team of 5-7 people.

Metrics from the peer review process can be used to help guide these risk based decisions. For example, as the metrics in Figure 5 illustrate, an organization determined that their most efficient inspections (defects found per hour) happen with 4-person inspection teams and their most effective inspections (total defects found) happen with 6-person inspection teams. This organization then used 6-person inspection teams for their inspections for work products with very high risk exposure and 4-person inspection teams on work products with lower risk exposure.




[based on Potter-01]

**Figure 5: Optimum Number of Participant Metrics**

Another risk-based peer review decision is **peer review sufficiency** – when to stop peer reviewing. After defects are found and fixed or improvements are made as a result of holding a peer review, should the work product be peer reviewed again? There is the possibility that not all the defects were found the first time or that new defects were introduced when changes were made. Risk-based peer reviews embrace the "Law of Diminishing Return." When a software work product is first peer reviewed, many of the defects that exist in the product are discovered with little effort. As additional peer reviews are held, the probability of discovering any additional defects starts to decrease. At some point, the return-on-investment (value-add) to discover those last few defects is outweighed by the cost of additional peer reviews. As illustrated in Figure 6, the probability of undiscovered defects still existing in the product and the potential impact associated with those defects must be balanced against the cost of performing additional peer reviews and the benefit of those reviews (just because an additional peer review is held, does not mean that additional defects will be found).


- Probability of undiscovered defects
- Potential impact associated with the defects
- Cost of additional peer review
- Benefits of more peer review

**Figure 6: Peer Review Sufficiency**

## Conclusions

Industry experience shows that peer reviews, especially inspections, are beneficial in improving the quality of our software work products. However, most projects have limited resources and peer reviews are just one of many activities that have to be accomplished within those limitations. By using a risk-based approach to selecting the type of peer reviews we hold, their level of formality, the number of participants and the peer review sufficiency needed for each work product, we can focus our resource investments where we can receive the highest return in improved quality and avoid overkill by expending resources where they are not value-added.

## References


Gilb-93 Tom Gilb, *Software Inspections*, Addison-Wesley, Wokingham, England, 1993.

IEEE-610 IEEE Standards Software Engineering, *IEEE Standard Glossary of Software Engineering Terminology, IEEE Std. 610.12-1990 (R2002),* The Institute of Electrical and Electronics Engineers, 1990.

Potter-01 Neil Potter and Mary Sakry, *Inspection Moderator Training*, The Process Group, 2001.

Westfall-08 Linda Westfall, *The CSQE Handbook*, Quality Press, 2008.

Wieger-02 Karl Wieger, *Peer Reviews in Software*, Addison-Wesley, Boston, 2002.