

Software review

A **software review** is "a process or meeting during which a software product is examined by a project personnel, managers, users, customers, user representatives, or other interested parties for comment or approval".^[1]

In this context, the term "software product" means "any technical document or partial document, produced as a deliverable of a software development activity", and may include documents such as contracts, project plans and budgets, requirements documents, specifications, designs, source code, user documentation, support and maintenance documentation, test plans, test specifications, standards, and any other type of specialist work product.

Contents

[**Varieties of software review**](#)

[**Different types of peer reviews**](#)

[**Formal versus informal reviews**](#)

[**IEEE 1028 generic process for formal reviews**](#)

[**Value of reviews**](#)

[**See also**](#)

[**References**](#)

Varieties of software review

Software reviews may be divided into three categories:

- [Software peer reviews](#) are conducted by one or more colleagues of the author, to evaluate the technical content and/or quality of the work.^[2]
- [Software management reviews](#) are conducted by management representatives to evaluate the status of work done and to make decisions regarding downstream activities.
- [Software audit reviews](#) are conducted by personnel external to the software project, to evaluate [compliance](#) with specifications, standards, contractual agreements, or other criteria.

Different types of peer reviews

- [Code review](#) is systematic examination (often as [peer review](#)) of computer source code.
- [Pair programming](#) is a type of code review where two persons develop code together at the same workstation.
- [Inspection](#) is a very formal type of peer review where the reviewers are following a well-defined process to find defects.
- [Walkthrough](#) is a form of peer review where the author leads members of the development team and other interested parties go through a software product and the participants ask questions and make comments about defects.

- Technical review is a form of peer review in which a team of qualified personnel examines the suitability of the software product for its intended use and identifies discrepancies from specifications and standards.

Formal versus informal reviews

"Formality" identifies the degree to which an activity is governed by agreed (written) rules. Software review processes exist across a spectrum of formality, with relatively unstructured activities such as "buddy checking" towards one end of the spectrum, and more formal approaches such as walkthroughs, technical reviews, and software inspections, at the other. IEEE Std. 1028-1997 defines formal structures, roles and processes for each of the last three ("formal peer reviews"), together with software audits.^[1] IEEE 1028-1997 was succeeded by IEEE 1028-2008.^[3]

Research studies tend to support the conclusion that formal reviews greatly outperform informal reviews in cost-effectiveness. Informal reviews may often be unnecessarily expensive (because of time-wasting through lack of focus) and frequently provide a sense of security which is quite unjustified by the relatively small number of real defects found and repaired.

IEEE 1028 generic process for formal reviews

IEEE Std 1028 defines a common set of activities for "formal" reviews (with some variations, especially for software audit). The sequence of activities is largely based on the software inspection process originally developed at IBM by Michael Fagan.^[4] Differing types of review may apply this structure with varying degrees of rigour, but all activities are mandatory for inspection:

- **0. [Entry evaluation]:** The review leader uses a standard checklist of entry criteria to ensure that optimum conditions exist for a successful review.
- **1. Management preparation:** Responsible management ensure that the review will be appropriately resourced with staff, time, materials and tools, and will be conducted according to policies, standards or other relevant criteria.
- **2. Planning the review:** The review leader identifies or confirms the objectives of the review, organises a team of reviewers and ensures that the team is equipped with all necessary resources for conducting the review.
- **3. Overview of review procedures:** The review leader, or some other qualified person, ensures (at a meeting if necessary) that all reviewers understand the review goals, the review procedures, the materials available to them and the procedures for conducting the review.
- **4. [Individual] Preparation:** The reviewers individually prepare for group examination of the work under review, by examining it carefully for "anomalies" (potential defects), the nature of which will vary with the type of review and its goals.
- **5. [Group] Examination:** The reviewers meet at a planned time to pool the results of their preparation activity and arrive at a consensus regarding the status of the document (or activity) being reviewed.
- **6. Rework/follow-up:** The author of the work product (or other assigned person) undertakes whatever actions are necessary to repair defects or otherwise satisfy the requirements agreed to at the examination meeting. The review leader verifies that all action items are closed.
- **7. [Exit evaluation]:** The review leader verifies that all activities necessary for successful review have been accomplished and that all outputs appropriate to the type of review have been finalised.

Value of reviews

The most obvious value of software reviews (especially formal reviews) is that they can identify issues earlier and more cheaply than they would be identified by testing or by field use (the "defect detection process"). The cost to find and fix a defect by a well-conducted review may be one or two orders of magnitude less than when the same defect is found by test execution or in the field.

A second, but ultimately more important, value of software reviews is that they can be used to train technical authors in the development of extremely low-defect documents, and also to identify and remove process inadequacies that encourage defects (the "defect prevention process").

This is particularly the case for peer reviews if they are conducted early and often, on samples of work, rather than waiting until the work has been completed. Early and frequent reviews of small work samples can identify systematic errors in the author's work processes, which can be corrected before further faulty work is done. This improvement in author skills can dramatically reduce the time it takes to develop a high-quality technical document and dramatically decrease the error-rate in using the document in downstream processes.

As a general principle, the earlier a technical document is produced, the greater will be the impact of its defects on any downstream activities and their work products. Accordingly, greatest value will accrue from early reviews of documents such as marketing plans, contracts, project plans and schedules and requirements specifications. Researchers and practitioners have shown the effectiveness of reviewing process in finding bugs and security issues.^[5]

See also

- Software quality
- List of software development philosophies

References

1. IEEE Std . 1028-1997, "IEEE Standard for Software Reviews", clause 3.5
2. Wiegers, Karl E. (2001). *Peer Reviews in Software: A Practical Guide* (<https://books.google.com/books?id=d7BQAAAAMAAJ>). Addison-Wesley. p. 14. ISBN 0201734850.
3. "IEEE Standard for Software Reviews and Audits" (<https://ieeexplore.ieee.org/document/4601584>). *IEEE STD 1028-2008*: 1–53. 2008-08-15 [2008]. doi:10.1109/IEEESTD.2008.4601584 (<https://doi.org/10.1109%2FIEEESTD.2008.4601584>). ISBN 978-0-7381-5768-9.
4. Fagan, Michael E: "Design and Code Inspections to Reduce Errors in Program Development", *IBM Systems Journal*, Vol. 15, No. 3, 1976; "Inspecting Software Designs and Code", *Datamation*, October 1977; "Advances In Software Inspections", *IEEE Transactions on Software Engineering*, Vol. 12, No. 7, July 1986
5. Charles P.Pfleeger, Shari Lawrence Pfleeger. *Security in Computing*. Fourth edition. ISBN 0-13-239077-9

In 2020 the first honest software review platform was launched, <https://Tekpon.com> to convince more and more platforms to generate real reviews from visitors, not paid ones. If we accept fake reviews, we will buy products we don't want

This page was last edited on 7 September 2021, at 12:28 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.