

The importance of code review as a learning and quality improvement tool for software development.

A literature review about the benefits of source code peer reviews.

Table of contents

Table of contents	1
Abstract	2
Introduction	2
Benefits of peer reviews	2
Focus on best practices	2
Propel teamwork	2
Knowledge transfer	2
Highly effective for bug detection	2
Saves time and money	3
Improves software security	3
Compliant with quality standards	3
Conclusion	3
Bibliography	4
Other references	5

Abstract

In this work, we review the literature about benefits of code review for quality improvement, cost and risk reduction, and minimization of impact of talent rotation. Also, we will share a short list of the standards that contain “peer review” or “code review” as a practice in software development processes.

Introduction

Code review is a well known tool for the software development industry. However, as quoted in K. E. Wiegers' text ¹:

“...many organizations struggle to implement an effective review program. Many of the barriers to successful peer reviews are social and cultural in nature, not technical.”

In order to overcome these barriers, the advantages of peer review must be known not only as a tool for the organization that wishes to implement it, but as a continuous improvement training activity for both, the reviewers and the authors of the code. The received reviews shouldn't be interpreted as defects or flaws in the author's code, but as possible improvements or suggestions, offered by the reviewer. In the same way, the reviewer shouldn't perceive them as work done for the benefit of the author, but, time that saves the team in the future, for example, identifying and correcting bugs or avoiding remaking features ².

To break down these barriers, a more in-depth analysis of the benefits of this process is presented below.

Benefits of peer reviews

Focus on best practices

Developers which already know that his or her code is going to be reviewed are motivated to solve in the most correct way, each and every one of the features of their implementation, this being a source of pride, and thus, improving individual best practices ³.

Propel teamwork

Reviews sometimes create conflict between the author and the reviewer ⁴, this dispute should be treated as a suggestion and not as an attack, however, it can help the team communication and its ability to solve any type of disagreement.

Knowledge transfer

When team members read and comment each other colleague's code, they improve the overall team knowledge of the project and increase the understanding of the related technologies, effectively preparing them for tackling the next tasks within the same or similar area of the solution.

Highly effective for bug detection

According to Steve McConnell, in his well-known work “Code Complete: A Practical Handbook of Software Construction” ⁵:

“...the average defect detection rate is only 25 percent for unit testing, 35 percent for function testing, and 45 percent for integration testing. In contrast, the average effectiveness of design and code inspections are 55 and 60 percent.”

Saves time and money

High quality code which is more readable and maintainable, hence reduces the time and cost of next development stages. Through code review, de facto common style guides and good practices are established, which minimizes the impact of talent rotation on the development team, thus avoiding the creation of critical developers ² in certain areas or functionalities, and thus granting greater freedom to request vacations or working in different tasks.

Improves software security

With peer review, each change or “commit” applies the knowledge of the author and all the reviewers. If at least one of the participants has security knowledge, or has heard of a new “exploit” that may affect the current code, the code will be modified to avoid these problems and all participants will keep part of that knowledge for similar future problems ⁴, thus improving the security of the project “*passively*”.

Compliant with quality standards

Due to its benefits, peer review is part of the certification process in multiple well-known standards of quality assurance in software development.

Those directly linked to peer review and code quality are ISO / IEC 20246 ² and CMMI-DEV ³ in the "Verification" section. To continue, ISO / IEC 25041 ⁹, mentions code review as an evaluation tool, ISO / IEC 14598 ¹⁰ covers code review for developers and it's used to apply and certify with ISO / IEC 9126 ¹¹.

Conclusion

Code review is one of the most useful and effective tools to become a high quality development team, certifiable by standards. Adding code review to the internal processes of the organization contributes to creating more reliable products and reducing development risks. In addition, not only it benefits the organization and the team, but also provides a continuous improvement tool for individuals.

Multiple authors ^{1,3,5,7} propose to establish peer reviews and style guides, so that the participants know what to look for, or what to avoid, in their reviews and authorship processes. In addition, providing a set of procedures and tips for reviews and / or authorship can facilitate the integration of the process in the current workflow.

Successfully implementing a culture which embraces the awareness of source code quality and peer review within the organization can be a major challenge ¹ sometimes due to developers' lack of understanding of the benefits of these practices.

As a result, currently there are multiple tools available which facilitate the code review practice. Among well-known version control services such as Github ¹², Bitbucket ¹³ or Gitlab ¹⁴, there are others like Gerrit ¹⁵ or CodeStriker ¹⁶. In addition, there are reputation management and gamification tools for software development, such as BrightByte ¹⁷, which promotes the code review practice, making it fun and a pride for developers, and thereby helping to integrate code quality awareness into the organization's culture.

Bibliography

- ¹Karl E. Wiegers, “Humanizing Peer Reviews”, https://www.processimpact.com/articles/humanizing_reviews.pdf
- ²Dan Radigan, “Why code reviews matter (and actually save time!)”, <https://www.atlassian.com/agile/software-development/code-reviews>
- ³Robert F, “Code Review Best Practices”, <https://medium.com/palantir/code-review-best-practices-19e02780015f>
- ⁴Derek Prior, “Implementing a Strong Code-Review Culture”, RailsConf 2015
<https://www.youtube.com/watch?v=PJjmw9TRB7s>
- ⁵Steve McConnell, “Code Complete: A Practical Handbook of Software Construction”
- ⁶Bruce Johnson, “Lessons From Google: How Code Reviews Build Company Culture”,
<https://blog.fullstory.com/what-we-learned-from-google-code-reviews-arent-just-for-catching-bugs/>
- ⁷ISO/IEC 20246, “Software and systems engineering – Work product reviews”,
<https://www.sis.se/api/document/preview/921568/>
- ⁸CMMI-DEV, “Mejora de los procesos para el desarrollo de mejores productos y servicios”,
<https://cmminstitute.com/getattachment/4439387f-28aa-4f3a-8f2b-a0cc5b449e47/attachment.aspx>
- ⁹ISO/IEC 25041, “Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE)”, <https://www.iso.org/obp/ui/#iso:std:iso-iec:25041:ed-1:v1:en>
- ¹⁰ISO/IEC 14598, “Tecnología de la información. Evaluación del producto software”,
<https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0034121>
- ¹¹ISO/IEC 9126, “Ingeniería del software. Calidad del producto software”,
<https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma/?c=N0032555>
- ¹² Github, <https://github.com/>
- ¹³ Bitbucket, <https://bitbucket.org/>
- ¹⁴ Gitlab, <https://about.gitlab.com/>
- ¹⁵ Gerrit, <https://www.gerritcodereview.com/>
- ¹⁶ CodeStriker, <http://codestriker.sourceforge.net/>
- ¹⁷ BrightByte, <http://brightbyteapp.com/>

Other references

- Jeff Atwood, "Code Reviews: Just Do It", <https://blog.codinghorror.com/code-reviews-just-do-it/>
- Smart Bear, "10 tips to guide you toward effective peer code review", <https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/>
- Madalin Ilie, "Code review guidelines", <https://www.codeproject.com/Articles/524235/Codeplusreviewplusguidelines>
- Google-eng, "Google's Engineering Practices documentation", <https://google.github.io/eng-practices/review/reviewer/>
- Perforce, "Coding Standards For Quality and Compliance", <https://www.perforce.com/resources/qac/coding-standards>