

Peer Review in Software Development: A Survey

Aylin GÜZEL¹, Özlem AKTAŞ² and Kökten Ulaş BİRANT³

^{1,2,3} Dokuz Eylül University, Computer Engineering Department Tınaztepe Campus, Buca İzmir Turkey

¹aylin.guzel@ogr.deu.edu.tr, ²ozlem@cs.deu.edu.tr, ³ulas@cs.deu.edu.tr

ABSTRACT

Analysis, decision making, detecting defects and correcting mistakes is important in software development process. Defects in the code reduce the quality of the software. Peer review is simple but has a huge impact on software quality. Also, peer review destroy defects early, easily, low-cost and efficiently. Code defects must be examined for better quality, high-performance, low-cost, re-use, modification and easy development of software. In this paper, the questions of what peer review is, why it is necessary for software development process and importance of peer review are told briefly by investigating in the literature.

Keywords: *Peer review, Quality of feedback, Software, Open source software.*

1. INTRODUCTION

Software review is a process or a kind of meeting types during which a software product (document, code, application etc.) is examined by a team members, managers, users, customers [1]. A code review reveals directly the location of a bug, while testing requires a debugging step to locate the origin of a bug. Reviews offer to detect and correct errors/defects early in the software development life cycle. Also, there are many types of software review such as software peer review, software management review, software audit review. Basically, technical reviews such as desk-check, walkthroughs, inspections, peer reviews, audits examine work products of the software project (code, requirement specifications, software design documents, test documentation, user documentation, procedures) for validation, verification and quality assurance purposes. Also, management reviews determine adequacy and monitor progresses or inconsistencies against plans, schedules and requirements.

Peer review aims for better software, high-performance, low-cost, re-use, better modification, easy and effective usage, easy development of software, high code readability and better quality. A peer review is an assessment of a product conducted by its author and one

or more colleagues informally. Software is written by human beings. Therefore, software is riddled with mistakes. In software development process, source code is evaluated by programmers, a review leader, the author and specialized professionals. Other people such as supervisors or managers do not participate in software peer review process.

Peer review has different types which are code review, pair programming, inspection, walkthrough, technical review, the email thread, over the shoulder. Code review is effective examination of computer source code. In the software development process, code review is a stage in which the authors of code, peer reviewers, and perhaps quality assurance testers get together to review code. Finding and correcting bugs at this stage is generally inexpensive. In addition, this process inclined to reduce the more expensive process of handling, locating, and fixing bugs during later stages of development or after programs are delivered to users [2].

Pair programming is one of the Extreme Programming (XP) and a kind of peer review in which two developers work on the same machine with their own keyboards and mouse. At any given time one is driver and the other navigator. The roles switch either every hour or whenever they determined. The driver codes, the navigator is reading, checking, spell-checking and testing the code, while thinking through problems and where to go next. If the driver hits a problem, there are two people to find a solution, and one of the two usually has a good idea.

Advantages of pair programming are two people have differing specialties; these skills are transferred amongst themselves. Developers talk and share ideas in the review process in an easy and helpful way. Also, developers are fully aware of the code, how it works, and why it was done in that way. Software is coded better than one developer working alone, as there was somebody watching [3].

Inspection is a very formal type of peer review where the reviewers are following a well-defined process to find defects [1]. The Formal Inspection Process is shown in Figure 1.



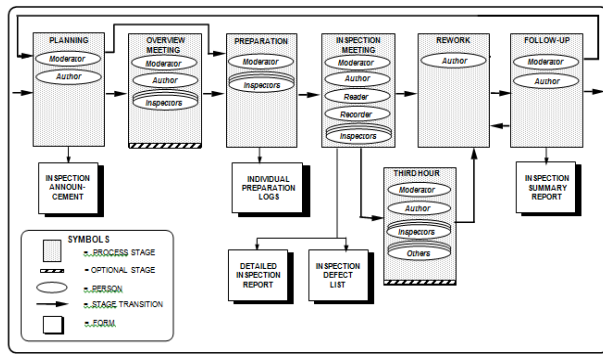


Fig. 1. The formal inspection process

Walkthrough is a form of peer review where the author leads members of the development team and other interested parties through a software product and the participants ask questions and make comments about defects [1]. The author reads the code line by line explaining what it does, reviewers listen and ask questions. In addition, walkthrough is a type of technical review where the producer of the reviewed material serves as the review leader and actually guides the progression of the review. In the case of code walkthrough, test inputs may be selected and review participants then literally walk through the design or code and checklist and preparation steps may be eliminated. The walkthrough process is shown in Figure 2.

Task	Responsible
1. Select review participants, obtain their agreement to participate, and schedule a walkthrough meeting.	Author
2. Distribute work product to reviewers prior to the meeting.	Author
3. Describe the work product to the reviewers during the meeting in any appropriate way. Lead discussion on the topics of interest or concerns about the work product.	Author
4. Present comments, possible defects, and improvement suggestions to the author.	Reviewers
5. Based on reviewer comments, perform any necessary rework of the work product.	Author

Fig. 2. The tasks of walkthrough.

An inspection is a more formal process than a walkthrough used to collect metrics or statistics about the software process. In other words, walkthrough is a more informal version of an inspection.

In a peer desk-check, only one person besides the author examines the work product. A peer desk-check typically is an informal review, although the reviewer could employ defect checklists and specific analysis methods to increase effectiveness [4].

A way of testing the logic of programs is to carry out a desk-check, that is executed the statements of the algorithm yourself on a sample data set. Desk-check process is respectively as it follows: product document, plan the desk-check, prepare checklist as optional, send documents to reviewers, review the product document,

conduct meeting if needed, edit document and finally, complete the review process.

In “over the shoulder process”, one developer looks over the author's shoulder while s/he is writing the code. This is the most common, informal and easiest of code review. Typically, the author "drives" the review by sitting at the keyboard and mouse, opening various files, pointing out the changes and explaining what he did. The author can present the changes using various tools, even go back and forward between changes and other files in the project. If the reviewer sees something incorrect, s/he can fix it immediately.

The most obvious advantage of over the shoulder review is simplicity in execution. Anyone can do it in any time without training. It can also be performed whenever you need it most, especially complicated change or an alteration. The over the shoulder process is shown in Figure 3.

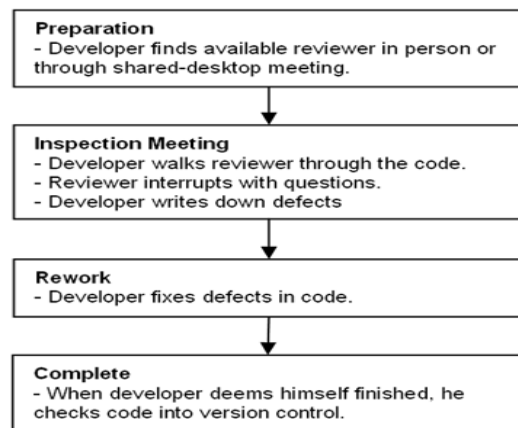


Fig. 3. The process of over the shoulder review.

In “email thread”, a piece of code is ready for review, the file is sent around to the appropriate colleagues via email to review as soon as their workflow permits. While this approach can certainly be more flexible and adaptive than more traditional techniques, an email thread of suggestions and differing opinions tends to get complicated fast, leaving the original coder on his/her own to sort through it all [5].

“Tool-assisted code review” is a kind of peer review that authors and reviewers use software tools, informal such as pastebins and IRC, or specialized tools designed for peer code review [5].

An audit is an independent evaluation of conformance of software products and processes to applicable regulations, standards, plans, and procedures and audit is a formally organized activity, with participants having specific roles, such as lead auditor, other auditors, a recorder, an initiator, and a representative of the audited organization. Audits can occur on almost any product at any stage of the development or maintenance processes.

2. ADVANTAGES AND DISADVANTAGES OF PERFORMING PEER REVIEWS

Performing software peer reviews has the following advantages:

- The peer review process doesn't involve management participation so the opportunity to work occurs in a friendly working area.
- In peer review process, peer reviewers promote software that is easy to read and maintain.
- Bugs are caught early.
- The peer reviewer process is cheaper and easier to fix than software testing activities.
- Peer reviews improve quality of product such as document, code, application by removing defects and also reduce cost.
- Improved team efficiency is a side-effect of peer reviews by improving team communication.
- Performing software peer reviews has the following disadvantages:
- They can require resources that are needed for other projects.
- They can be mired down in disputes over personal programming styles and preferences.
- They often require follow up action to make sure that the software has been modified in accordance with peer suggestions.

3. PREVIOUS STUDIES IN PEER REVIEW

Bride Mallon [6] had studied about peer review process for games and software in 2008. For this purpose, Mallon emphasized that a peer review process for assessing the contribution of artifacts, such as games and software was suggested. Games and software produced as research output by academics are tend to be accredited within their institution through discussion of the artifact, rather than directly. Mallon complained about academic paper evaluation criteria and not to get sufficient career credits from academic institutions for producing games, simulations, and software in his paper.

Consequently, he claimed that academics were tend to approach games and software simulations by producing written materials (such as design guidelines or evaluations or analysis of the development process) around the artifact, rather than to seek greater involvement in creation of the artifacts themselves. A key factor influencing this imbalance is a well-

established process for accrediting written materials, for judging their contribution and quality, existing the peer review process whereas a comparable process for judging and accrediting the artifacts themselves does not.

According to Mallon, the benefits of a peer review process for software and games:

- Credit for these “publications” may be sought by academics, the artifact having been judged and evaluated by peers.
- Offering standards for games and software evaluation.
- Furnishing accreditation of products for industry developers.
- Giving formative feedback to authors or developers for improving products.
- Supplying publication outlets to serve as repositories for knowledge.
- Arun Kumar [7] investigated anatomy and physiology of peer review process, and questioned whether it can be unbiased, in his research. He defined the peer review as follows:
- Peer review is the process of evaluating someone’s work by at least one or more respective subject experts, which has several formats and is currently applied in several domains.

Kumar emphasized that the main aim of the peer review was to approve scientific quality and credibility of the work, the process of peer review can only be valid if it is 100% unbiased and this has emerged as a major limitation of the peer-review process. He examined some question’s answers such as how can we ensure the peer-review process was unbiased, who peer-reviews the peer reviewer, etc. He told that biased peer review process was scientifically and ethically wrong. He suggested that in developing a software or an application, unbiased peer review process must be used, also he suggested to remove biasness in this approach. As a result, Kumar said that publications could be made in a public platform (YouTube, Tweets, Facebook, LinkedIn). History of scientific publication process is shown in Figure 4.

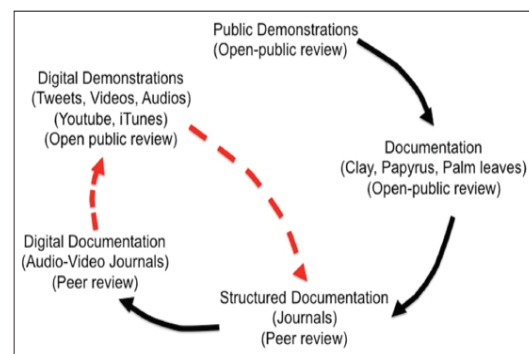


Fig. 4. History of scientific publication process.

Peter Rigby and his friends [8] studied on open source software (OSS). They examined more than 100,000 peer reviews in OSS case studies of the Apache Httpd server, Subversion, Linux, FreeBSD, KDE, and Gnome. Also, the projects broadcast changes dynamically to the development team usually on a mailing list and reviewers self-select changes they're interested in and competent to review. Developers manage what can be an overwhelming broadcast of information by relying on simple email filters, descriptive email subjects, and detailed change logs. The change logs represent the OSS project's heart, through which developers maintain a conceptual understanding of the whole system and participate in the threaded email discussions and reviews for which they have the required expertise. The spectrum of peer review techniques, from formal inspection to minimal process open source software review is shown in Figure 5.

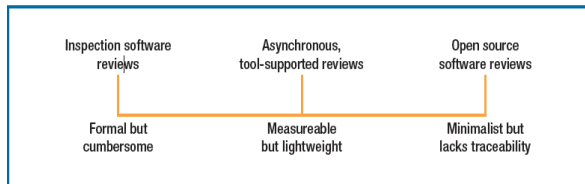


Fig. 5. The spectrum of peer review techniques.

The authors explained that asynchronous reviews supported team discussions of defect solutions and found the same number of defects as collocated meetings in less time. They also enabled developers and passive listeners to learn from the discussion. They said that the earlier a defect was found, the better for frequent reviews. OSS developers conduct all-but continuous, asynchronous reviews that function as a form of asynchronous pair programming. Also, most OSS peer reviews begin within hours of completing a change, and the full review discussion which involves multiple exchanges usually takes one to two days. Indeed, the feedback cycle is so fast.

Incremental Review should be of changes that are small, independent, and complete for the authors. Invested experts and code developers should conduct reviews because they already understand the context in which a change is being made for invested experienced reviewers.

Empower expert reviewers are expert developers who can let self-select changes they're interested in and competent to review. Reviewer types and their costs, investment level in the code, review quality, and amount of knowledge transfer and community development that occurs during the review is shown in Figure 6.

Reviewer type	Cost	Investment	Quality
Independent reviewer	Very high	Low	Medium
Pair programming	Very high	Very high	High
Codeloper reviewer	High	High	High
Regular incremental reviewer	Medium	Medium	Medium
Reviewer as bug fixer	Low	Medium	Low

Fig. 6. The spectrum of peer review types.

Laura Stemmler and Keith Collier [9] suggested a new mission called Rubriq for tools, services, and software to improve peer review process in their paper. Rubriq was on a mission to put lost time back into research. They estimated that 15 million hours are lost each year to redundant peer review as papers get rejected and flow down the journal prestige pyramid. Rubriq uses an author-pays model to facilitate fast, independent, and standardized peer review performed by three academic peers who are financially compensated for their efforts. This service was designed to improve journal selection, supplement editorial reviews, and make peer review more portable between journals.

4. CONCLUSION

Software review is a process of examining the code by a team members, managers, users, customers. This process is very important in the software development. Peer review is one of the processes that increase software quality. This paper reveals what is peer review, how do we use it, advantages and disadvantages of peer review process and peer review types. It was seen that, most of the papers published were not about software peer review, they were about peer review in education, academic world or medicine. The value of peer review in software should not be underestimated, and absolutely needed attention must be given.

REFERENCES

- [1] Software review, https://en.wikipedia.org/wiki/Software_review, [Date Accessed: 10 December 2015]
- [2] What is code review?, <http://searchsoftwarequality.techtarget.com/definition/code-review>, [Date Accessed: 10 December 2015]
- [3] Pair Programming vs. Code Reviews, <http://blog.codinghorror.com/pair-programming-vs-code-reviews/>, [Date Accessed: 10 December 2015]
- [4] Seven Truths About Peer Reviews, http://www.cs.pomona.edu/~markk/cs121.s07/supp/seven_truths.html, [Date Accessed: 11 December 2015]
- [5] What is Code Review?, <http://smartbear.com/learn/code-review/what-is-code-review/>, [Date Accessed: 10 December 2015]
- [6] Mallon, B., "A Peer Review Process for Games and Software", *New Directions for Evaluation*, Vol. 118, 2008, pp. 37-45.

- [7] Kumar, A., "Anatomy and physiology of peer review process, can it be unbiased?", Journal of Natural Science, Vol. 5, February 2008, pp. 1-2.
- [8] Rigby, P., Cleary, B., Painchaud, F., German, D., Storey, M., "Contemporary Peer Review in Action: Lessons from Open Source Development", IEEE Software, December 2012, pp. 56-61.
- [9] Stemmele L., Collier, K., "RUBRIQ: tools, services, and software to improve peer review,", Learned Publishing, Vol. 26, October 2013, pp. 265-268.

AUTHOR PROFILES:

Aylin GÜZEL was graduated from Bilecik Şeyh Edebali University Department of Computer Engineering in 2013 in degree of BSc. She had lessons from Anadolu University Computer Engineering Department as Farabi Programme student in 2011-2012 Fall and Spring semesters. She is a student in Dokuz Eylül University Department of Computer Engineering MSc. Programme.

Özlem AKTAŞ was graduated from Dokuz Eylül University Department of Computer Engineering Department in 2003 in degree of BSc. She was graduated from the same department in 2005 in a degree of MSc. and in 2010 as Ph.D. She has been working as Assistant Professor in the Dokuz Eylül University Department of Computer Engineering Department since 2010.

Kökten Ulaş BİRANT was graduated from Dokuz Eylül University Department of Computer Engineering Department in 2000 in degree of BSc. He was graduated from the same department in 2002 in a degree of MSc. and in 2010 as Ph.D. He was working as research assistant in Dokuz Eylül University Department of Computer Engineering Department in 2000-2010, and he has been working as Assistant Professor in the same department since 2010.

