

SE 811: Software Maintenance

Toukir Ahammed

Inspecting Changes

Software Inspection and Modern Code Review Practices

Code Inspections

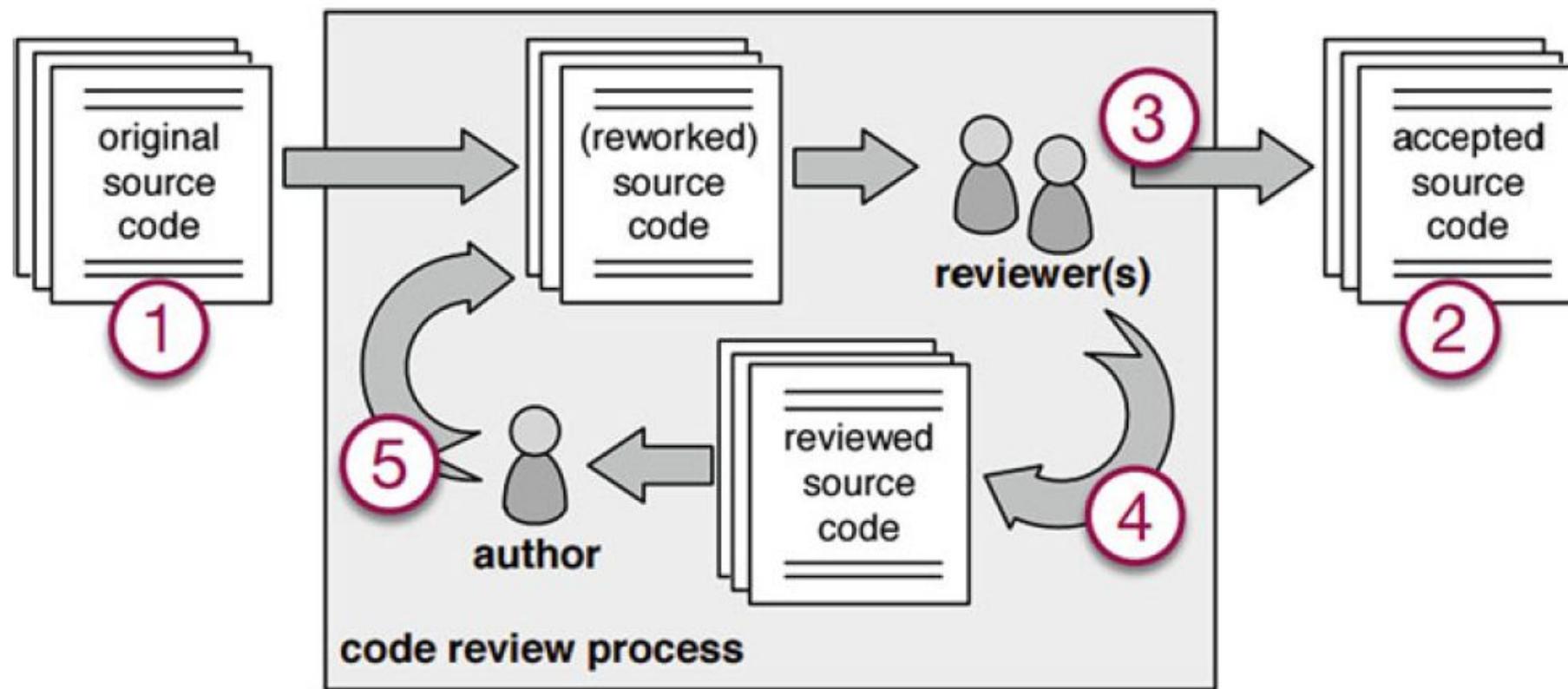
- To improve software quality during software evolution, developers often perform *code reviews* to manually examine software changes.
- Code inspections are performed at the end of major software development phases, with the aim of finding overlooked defects before moving to the next phase.

Traditional Code Inspections

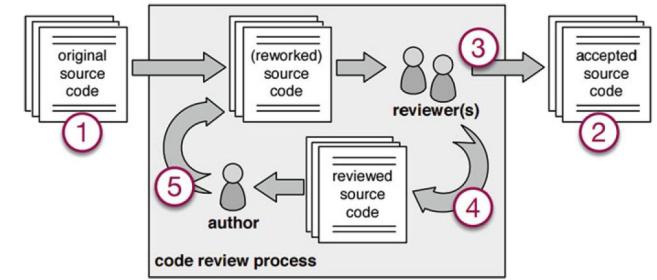
- Software artifacts are circulated a few days in advance and then reviewed and discussed in a series of meetings.
- The review meetings include the author of an artifact, other developers to assess the artifact, a meeting chair to moderate the discussion, and a secretary to record the discussion.
- Over the years, code inspections have been proved a valuable method to improve software quality.
- However, the cumbersome and time-consuming nature of this process hinders its universal adoption in practice.

Modern Code Reviews (MCR)

- To avoid the inefficiencies in code inspections, most open-source and industrial projects adopt a lightweight, flexible code review process, which we refer to as *modern code reviews*.



Modern Code Reviews (MCR)



- The *author* first submits the *original source code* for review.
- The *reviewers* then decide whether the submitted code meets the quality acceptance criteria.
- If not, reviewers can annotate the source code with review comments and send back the *reviewed source code*.
- The author then revises the code to address reviewers' comments and send it back for further reviews.
- This process continues till all reviewers accept the revised code.

Modern Code Reviews (MCR)

- In contrast to formal code inspections, modern code reviews occur more regularly and informally on program changes.
- Modern code reviews can be described as “early, frequent reviews of small, independent, complete contributions conducted asynchronously by a potentially large, but actually small, group of self-selected experts.”

Modern Code Review Practices

Modern code reviews occur early, quickly, and frequently.

- Traditional code inspections happen after finishing a major software component and often last for several weeks.
- In contrast, modern code reviews happen more frequently and quickly when software changes are committed.
- For example, the Apache project has review intervals between a few hours to a day. Most reviews are picked up within a few hours among all projects, indicating that reviewers are regularly watching and performing code reviews

Modern Code Review Practices

Modern code reviews often examine small program changes.

- During code reviews, the size of software change varies.
- The change size in modern code reviews is much smaller than code inspections.
- Such small changes facilitate developers to constantly review changes and thus keep up-to-date with the activities of their peers.

Modern Code Review Practices

Modern code reviews are conducted by a small group of self-selected reviewers.

- In OSS projects, no reviews are assigned, and developers can select the changes of interest to review.
- Program changes and review discussions are broadcast to a large group of stakeholders, but only a small number of developers periodically participate in code reviews.
- In industrial projects, reviews are assigned in a mixed manner—the author adds a group of reviewer candidates and individuals from the group then select changes based on their interest and expertise.

Modern Code Review Practices

Modern code reviews are often tool-based.

- There is a clear trend toward utilizing review tools to support review tasks and communication.
- Previously code reviews in OSS projects were often email-based due to a lack of tool support.
- More recently, popular OSS hosting services such as GitHub and BitBucket have integrated lightweight review tools to assign reviewers, enter comments, and record discussions.
- Compared with email-based reviews and traditional software inspections, tool-based reviews provide the benefits of traceability.

Modern Code Review Practices

Although the initial purpose of code review is to find defects, recent studies find that the practices and actual outcomes are less about finding defects than expected.

- A study of code reviews at Microsoft found that only a small portion of review comments were related to defects, which were mainly about small, low-level logical issues.
- Rather, code review provides a spectrum of benefits to software teams, such as knowledge transfer, team awareness, and improved solutions with better practices and readability.