Mehul Ganjude 1001990551

**SOFTWARE ENGINEERING II: MGMT, MAIN AND QA**

**PROF. GIRIDHAR AMARAVADI**

**Automated Code Reviews (SDLC)**

**Preamble: -**

Automated code review is the act of automatically going through a source code to find ineffective or mediocre code using a preset set of standards. The automatic code review tools assist programmers in locating errors and spotting potential security flaws. When checking if the code complies with the company's requirements during the code review process, these tools typically present warning indications. A program that reviews code automatically can correct errors or instruct people on how to do so. Code review, also referred to as peer code review, is the deliberate and methodical gathering of one's fellow programmers to examine each other's code for errors. It has been repeatedly demonstrated that code review can speed up and simplify the software development process unlike few other techniques.

**Project author View: -**

The authors have discussed, Although, there are methods and technologies for peer code review, since software is written by humans, it is obvious that software contains errors, but it is less obvious why software developers frequently rely on manual or automated testing to vetting their code. Additionally, code reviews may help you save money by discovering the kinds of defects that could otherwise get past testing, production, and even your end users. Common Approaches to Code Reviews: -

**Pair Programming:**

With this method of building software components, developers work side by side on the same code while simultaneously verifying each other's work. Code reviews take place within the development process, making it a suitable approach for experienced developers to mentor young mentees.

**Tool-Assisted, Best Practice for Code review:**

compared to using software-based code review tools, there are browser-based or easily connect with numerous common IDE. Other features of certain solutions include the ability to examine and edit requirements papers as well as, more importantly, the ability to provide important use data that provide the audit trials and review metrics required for process improvement and compliance reporting. strategy must strike a balance between well specified procedures and a friendly, cooperative atmosphere.

**Certain Tips Author has mentioned for effective peer code review:**

**At a time, review a few lines of code**

In real life, an examination of 200–400 LoC over 60–90 minutes should uncover 70–90% of the defects. Therefore, if there were 10 flaws in the code, a thorough inspection would discover between 7 and 9 of them.

**Set objectives and record measurements**

Start establishing some measurements based on various standards. This will help you to measure how your code is progressing.

**Recommended procedures for code reviewers:**

* Inquiries, not assertions: An allegation may be inferred from a statement. If a statement is your only option, take these into account:
* An issue can have several fixes: Just because a piece of code differs from how you imagined it could be written doesn't make it bad.

Diagram

Description automatically generated

Source: <https://medium.com/globant/code-reviews-5a648ff7b971>

**Potential Generalized View:**

There are many methodologies are being followed by every company or individual to test the code, some are followed checklist or some user the tools. There is not a particular technique that has to stick to do test. The benefits of code review are code standard compliance, debugging, Higher software security and team cohesion. There is certain process must follow to review the code read less time, plan accordingly, follow checklist, Immediately Fix Defects Found.

There are certain benefits for web product owner like code review reduces risks, it detects bug at initial stage, it saves money, improve web security. Along with this it provides benefits to web product team like enforce complains of coding standards, it promotes sharing knowledge, help mentor to newer developers. Even when the developer is still coding, an automatic code review tool may swiftly find problems. Automated code review tools do not make human-like manual mistakes. They carry out perfect rule-based audits: If they were designed to recognize a specific fault, they would undoubtedly do so. Code reviews performed automatically are impartial. This makes the automated code review different than code reviews.

Using a static code analysis tool is the ideal way to handle the initial stage of evaluating the code quality, which may be a nuisance of enormous proportions. To get a thorough examination of the code, use plugins like SonarQube, NDepend, TFS, and FxCop. When experimenting with Visual studio, ReSharper is another excellent plugin. Utilize technologies like Crucible, Bitbucket, and TFS code and finish the review process to track the code review comments.

**Diagram, timeline

Description automatically generated Diagram

Description automatically generated**

Source:

* <https://medium.com/sciforce/what-is-code-review-881de5a273cb>
* <https://medium.com/cuelogic-technologies/code-review-process-best-practices-3eeecab26ded>
* <https://research.aimultiple.com/auto-code-review/>

**My Viewpoint:**

From the above articles it has proven that automated testing reduces manual task. It also helps to improve the performance and follows disciplinary approach. But to implement automated testing there are many factors need to be taken care. It reduces human task to debug code many times and cost is less. The only thing is managing the checklist that has been assigned to test the code and keep track of every phase when it is in its debugging level. It allows coders to review the code during their own time and that too in a distributed fashion. Additionally, code review promotes social recognition, which results in improved job outcomes. The ability to evaluate someone's work and receive comments is a great advantage since only skilled developers are doing evaluations. Moreover, code reviews call for the reviewer and reviewee (developer) to keep track of security domain updates and use that knowledge in real-world situations. Security thus joins the tasks that are given high importance. It also provides some organizational benefits like efficiency, code quality standardization, code quality predictability, team management and proactive.

Code reviews work best when they are included into a team's established procedures. Consider conducting a code review once all the code has been designed, the automated tests have run, and they have been successfully passed, regardless of whether you are utilizing a branching workflow. Merge the branches with the main code only after that. Companies can buy the tools to test the code or certain team will have high standards to test code, so third party team also can play an important role on this. Software development is crucially dependent on manual code reviews. They give the developer with a measure of responsibility by bringing in a variety of new viewpoints from specialists to spot logic flaws, validate the code works as intended, and confirm it does so, among other things. While a manual review focuses on logic and intent, an advanced automated code review may assist produce a safer and more effective application much more quickly than a manual review alone. You'll find more problems more quickly using an automated review, which will enhance the development process's correctness and speed.

In the Agile technique, code reviews are a crucial component of the software development life cycle (SDLC). Even though Scrum emphasizes iterative development, developers still need to complete the project before the deadline. In other words, employees must finish their work by the deadline listed on the burn-down schedule for their project. To accomplish this goal, they must reduce risks and mistakes and adhere to best practices. To achieve it, though, they need trustworthy, well-tested code. Code reviews therefore perform the most important part in the SDLC.