CS-499-Q2866 Computer Science Capstone

2-1 Journal: What Makes a Productive Code Review

11/10/2018

Richard S. Whittaker

# What is Code Review?

Code review is an efficient inspection of the code to find and correct vulnerabilities which can cause security flaws, memory leaks and buffer over-runs/flows. Code review is an audit of the source code to ensure the software or application was designed to protect it from security threats. Code reviews are not generally done by the developer instead done in a peer review group. The review is not just for finding security vulnerabilities within the code. Though this practice will help identify those security concerns but is convenient in finding different ways the program can be streamlined and simplified to be more efficient (Smartbear.com, 2018).

Throughout the code review, developers look for adherence to style and standards in following the accepted guidelines for coding styles. Constancy in naming of fields, variables, packages, etc and formatting for correctness and consistency within the code. They are used to evaluate test coverage, test readiness and availability to test the code. The review will also compare the code against genuine code principles and checked for readability, functionality and error handling. There are many tools that can do the code review for you but in many cases, it is the programmers looking at each line of code to determine if the code that worked with the current complier might not run with the other compliers. Normally it is the human that is catching more errors than many of the tools and programs designed to accomplish these reviews (Schindler, 2018).

# Why is it an important practice for computer science professionals?

Code reviews are an important practice for computer science professionals because it helps in finding errors, bugs, vulnerabilities early in the software development lifecycle, decreasing time and resources. Code review is more effective than standard software testing and helps find and identify code issues. The average effectiveness of design and code reviews are 55% to 60%. Unit test are not very effective and only 25% of the time when detecting errors, along with function and integration test only proved to be 35% to 45% accurate (ludovicianul, 2018).

# When and how does it occur?

Code review normally occurs in the early in the software development process. The group should review no more than 200 to 400 lines of code or about 60 to 90 minutes to be effective (Smartbear.com, 2018). The review should be done after the preliminary development stage and continue through testing and verification and especially after the unit test have completed.

# What are some code review best practices that you would advocate?

Code review best practices that I would advocate is to review less than 400 lines of code at a time. Establish roles and responsibilities for the developer and reviewer, set goals for the code review for example setting severity to find issues (high, medium, low) while focusing on the most severe severity first. Another practice, is to create a checklist throughout the process which will certify that all areas are covered during the code review.

# Optional: Are there any practices that may be currently uncommon that you believe would make code reviews more effective?

One of the most effective and uncommon practices is Test Driven Development (TDD). TDD is a software development process that the developer writes the test case prior to writing code for the application. TDD ensures the test case are written first before the actual implementation is written. This will help make sure the code is written with testability and coverage in mind while defining the “Not Started”, “Not Done”, “In Progress”, and “Completed. Though many organizations feel this is to costly and takes too much time from being productive. However, studies have shown that the cost to correct errors, bugs, vulnerabilities, etc. in a production state is 100 times greater than those correcting the issues during the design phase (Saini, 2018).

# References

Smartbear.com. (2018). *What Is Code Review?*. [online] Available at: https://smartbear.com/learn/code-review/what-is-code-review/ [Accessed 10 Nov. 2018].

Schindler, E. (2018). *What to Look for in a Code Review*. [online] CIO. Available at: https://www.cio.com/article/2431554/developer/what-to-look-for-in-a-code-review.html [Accessed 10 Nov. 2018].

ludovicianul (2018). *Code review guidelines - CodeProject*. [online] Codeproject.com. Available at: https://www.codeproject.com/Articles/524235/Codeplusreviewplusguidelines [Accessed 10 Nov. 2018].

Saini, D. (2018). *Use approvals to enforce test driven development, code reviews and integration testing in Rational Team Concert*. [online] Ibm.com. Available at: https://www.ibm.com/developerworks/rational/library/approvals-test-driven-development-code-reviews/index.html [Accessed 10 Nov. 2018].