**Assignment 6.3 Code Review**

*Becca Buechle*

*October 21, 2020*

Code review plays a very important role in the development world. It helps developers of all coding skill levels find errors before their code is pushed out live and left to cause possible errors there. When errors are found before the product goes live it can save the company a lot of time and money later trying to come up with a fix for something that has broken after a change has been made to it. Below I’ll go into detail on some tips to help make code review go much smoother and why it’s important to do.

Google did a case study of Code Reviews (2010) and found that breaking up code into smaller parts makes it much easier for a Senior developer to work through when doing code review. Randy Shoup mentioned starting a project and working on it for weeks and ended up having around three thousand lines of code. (Kim, G.) He then turned it into his review team, and it took them days to get through because there was just so much to loo through. (Kim, G.)

After deep diving into some research I found some of the best practices for doing code review. Tip one, when doing any sort of code review you should be reviewing two hundred to four hundred lines or less at a time. (SmartBear.) This is due to the human brain only being able to process so much at one time and when you have more than four hundred you will start to lose the ability to find small errors. (SmartBear.) SmartBear also mentioned when doing about two to four hundred lines it talks about one to two hours to do and the reviewer will catch about 70-90% of the errors. (SmartBear.) Tip two, when reviewing code, you may want to rush through to just get done but when reviewing any type of code, you’ll want to take your time. When reviewing with a limited amount of code like mentioned above and at a slow pace you should be able to review the most effective way. (SmartBear.) Tip four, when doing code review limit the amount of time to no more than two hours. (SmartBear.) If you’regoing to review code for longer break it up so that its not done all at one time. (SmartBear.) Doing this will ensure that you keep your performance up since studies have showed that it starts to drop off after 60 minutes. (SmartBear.) Tip five, make code annotations so that you can explain your reason for making a change to something. This can help in several ways when the person goes back to look at the changes that have been made it may teach them how to do something correctly in the future. (Cohen, J.) Adding annotations goes for both people though the reviewer and for the coder. (Cohen, J.) When the coder adds annotations, it can also help them see errors while they are doing through it before turning it over to the reviewer. (Cohen, J.)

To sum it up reviewing code is very important. When you go to do code review you should aim to do the work in short sitting periods so that your ability to find errors stays high. You also, want to review code with no more than four hundred line because anymore can send your brain into overload. Making notes in the code is also very important as it can teach someone something they didn’t know before. Overall, I believe that code review is a great tool for developers because it can help to catch some major bugs before they are released to the public. It can also save the company a lot of headaches by trying to fix something that has been released that doesn’t work.

**Bibliography**

Cohen, J. (2011, January 25). 11 proven practices for more effective, efficient peer code review. Retrieved October 21, 2020, from https://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). DevOps Handbook. Portland, OR: IT Revolution Press.

SmartBear. (2020). Best Practices for Code Review. Retrieved October 21, 2020, from https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/