Mod 2 Journal

By: Alex Roberts

Static testing is a method of testing software without even running it. It is usually implemented before the code is written and is planned out way ahead of time. This method finds the causes of failures instead of just finding the failure itself. It typically includes requirements or specification documents and testing the code without executing it. The way that this helps is because you are implementing this ahead of time so you can test it as you are working which makes it easier to figure out where the problem is instead of having to filter through the entire program. It removes defects and ensures that defects will not be in the final code.

Dynamic testing is similar to static and they both have the same goal but dynamic testing is carried out while executing the code and is usually done with the final product after all the coding is done. That is the main difference between the two. Static testing is the better option typically because you are more likely to get rid of the defects as you go and be able to find the issues faster. This reduces cost and time as the issues are much easier to fix if you catch them on the spot instead of having to go through every line of code and figure out what is triggering what. That takes so much more time and costs a lot more.

In conclusion, The best way to do this is to implement both. Use static testing before you even start writing the code and throughout to work out all the defects. Then at the end after all of the code is included in the program, use dynamic testing to tie it all together and work out the remaining errors which shouldn’t be as bad since you are testing throughout the entire process. In the end, if you follow this method and it passes all the reviews you should have a proper working program.