Week 7 task 2

Dynamic and static analyzers are both tools that are used to run tests on code, but they have many differences. in general static analyzers differ from dynamic in a way that static analysis examines source code while dynamic analysis looks at problems while running the code. So because dynamic analyzers also look at problems, they require much more time to run. Furthermore a static analysis can only be performed if the code and its mechanism doesn't change over time and the variables and inputs stay constant.

Both have their own place to use, but in certain cases one has an advantage over the other. For example Dynamic analyzers detect runtime issues like memory leaks, bottleneck problems and other bugs during the execution of the code while static analyzers don't. On the other hand, static analysers cover even the parts of the code that are not executed during the test while dynamic analyzers don't. Dynamic analyzers also cost much more than static analysers since they also look at the problems while running the code, which requires more resources. So in general dynamic analysers are better for code that needs to be runtime-tested and static analysis is good for checking code without running it and it doesn't look at the problems like dynamic analyzers.

One method of dynamic analysing is fuzzing. Fuzzing means testing programs by giving them random inputs. This is dynamic because the inputs keep changing, whereas static analyzers use constant inputs.

sources: https://digma.ai/static-analysis-vs-dynamic-analysis/