**Assignment questions:**

1. What is non-functional testing? give some examples of non-functional tests that are not mentioned in the lecture’s slides.
2. Is usability testing suitable for automation? give your insights.
3. Make a comparison between white box, black box, and gray box testing. Mention some testing techniques that go under each method, the suitable test cases for each method, and the skills a tester should have to conduct each test.
4. What is the difference between debugging testing and mutation testing techniques?

1. Non-functional testing refers to testing of software specifications, i.e. verifying that the product works, and how well it works. This will check the system’s non-functional parameters, such as usability, performance, security, etc.

Examples:

- Test if multiple or a large amount of users can work on the system simultaneously without the system being overloaded or failing.

- Test if the system can run on different operating systems and environments without any error.

- Test if the system can boot up and perform the required task in an acceptable amount of time.

2. Usability testing checks whether the system is easy for a group of users to learn, understand, and operate a particular software application. Usability testing requires random, gestural, and behavioral human inputs, which makes automation for this type of testing difficult or even impossible in some cases.

Human interactions are challenging to be converted to scripts. We need to make a lot of assumptions when automate this testing due to the complexity of human nature. We can create a script that predicts human these actions, but it's preferable to have people actually do the work in order to get an accurate image. In the end, the purpose of a usability test is to assess user behavior.

However, there are some usability testing tasks that could be automated. For example, we can record facial expressions of test subjects, and then feed them to an AI for analysis. It is not necessary to have a human to review all these recordings.

In conclusion, the majority of usability testings need are not suitable for automation. Nevertheless, whenever it is possible, we should automate the tasks. We should avoid making manual testing bottlenecks the testing process.

3.

|  |  |  |
| --- | --- | --- |
| **Black box** | **Gray box** | **White box** |
| Internal structure of the system is not visible to testers | Internal structure of the system is partially visible to testers | Internal structure of the system is visible to testers |
| Least comprehensive testing | Medium level of testing comprehensiveness | Most comprehensive testing |
| Testers do not need to understand system architecture | Testers need to have understanding on the programming level and on the high level system functions | Tester needs to know the software architecture and programming |
| Mimic a true external cyber attacks | Assess vulnerability against internal threats | Simulate an attack where the attackers have privileged access |
| Used in all test phases | Used in unit, integration, and regression phase | Used in unit, integration, and regression phase |
| Techniques:  Decision table testing  All-pairs testing  Equivalence partitioning  Error guessing | Techniques:  Matrix testing  Regression testing  Pattern testing  Orthogonal array testing | Techniques:  Control flow testing  Data flow testing  Branch testing |

4. Bebugging testong testing is a type of software testing in which known software bugs are added to the source code, and testers will find them. The proportion of known bugs added and not found gives an approximate indication of real bugs that remain undetected.

Mutation testing is a type of software testing in which small segments of the source code are changed, or “mutated”, to check if the test cases are able to find errors in source code.

Mutation testing differs in the sense that the program is modified by changing the code itself and several versions are created for the sole purpose of testing. On the other hand, in bebugging, the program is modified by adding bugs to the source code.