Assignment 3

Types of Manual Testing

**Types of manual testing**

Manual testing has many variations, with different types suited to different software and environments. Let’s take a look at some of the most commonly-used manual testing techniques.

**Acceptance Testing**

User Acceptance Testing (UAT) is performed by the client or end-user, to confirm that the software meets the agreed requirements. Sometimes called pre-production testing, it takes place during the final phase before releasing the product to market.

UAT is an example of functional testing and types of acceptance testing include Alpha (executed within the organization) and Beta (where the[application](https://www.globalapptesting.com/blog/perfect-app) is released to a limited market to generate user feedback).

A screenshot of a computer

Description automatically generated with low confidence

**Black Box Testing**

Also known as behavioral testing, this method aims to analyze an application’s functionality from the end-user’s perspective. The internal code structure is not visible during testing (hence the name “Black Box”), so testers are only aware of the inputs and expected outputs of the software.

Black Box Testing has several subdivisions, including functional testing for requirement compliance, smoke testing to assess basic functionality, and partitioning (dividing software into groups that are expected to exhibit similar behavior).

**Integration Testing**

Integration Testing is the process of testing an application with two or more integrating components. It is performed once the individual components have been unit-tested, and aims to identify problems with the interfaces and the interactions between them.

The two main methods are the Bottom-Up Approach (moving steadily from the bottom module to the top module) and Top-Down Approach (the opposite).

**System Testing**

System Testing means testing the system as a whole, once all its components have been unit-tested and integrated. It checks that the complete application works as intended, by comparing it against the original requirements.

Also called end-to-end testing, it typically involves installability testing (does the software install correctly?) and recovery testing (can the application recover from hardware crashes and network failures?).

**Unit Testing**

This is when the individual units or components of an application’s source code are tested, to make sure each function performs as expected. It is usually carried out by developers rather than engineers, as it requires detailed knowledge of the internal program design and code.

Also known as module testing or component testing, it simplifies the debugging system and helps to detect and protect against bugs in the future.

Diagram

Description automatically generated

[Image Source](https://fortegrp.com/the-importance-of-unit-testing/)

**White Box Testing**

Sometimes called transparent box testing or structural testing, this is a method of testing the internal structures or workings of an application. It is performed by the developer, who checks the software’s internal codes before passing it to a test engineer.

The main focus of White Box Testing is on strengthening security and improving the software’s design and usability. A combination of Black Box and White Box testing is known as Gray Box Testing.

**How manual testing is performed**

We’ve already seen that there are several types of manual testing, but the overall[testing process](https://www.globalapptesting.com/how-we-help/improve-product-quality) is common to all of them. Sometimes this will be carried out in-house, while some businesses prefer to engage an expert company to handle testing.

There are roughly six basic stages in performing manual testing:

**1.   Understand requirements**

The first thing testers need to do is to fully understand the project’s requirements. What does the client expect from the application? What problem is it aiming to solve for end-users? Testers must analyze all requirement documents in order to recognize the expected behavior of the software and exactly what needs to be tested.

**2.   Prepare test cases**

Once the requirements are understood, testers can draft test cases to cover various scenarios, such as what happens when a user enters an invalid password or how the software would cope with a crash. These test plans will set out a sequence for testing functionality and usability across the entire application, measured against expected results.

**3.   Review test cases**

It’s helpful to review the draft test cases with team leaders and the client, to check that they will cover all bases and make any amendments before commencing the execution. This will save time in the long run.

**4.  Execute test cases**

Manual testing can now be carried out, using any of the techniques listed in the previous section. As well as finding bugs, the aim is to identify potential pain points for users and loopholes that could be exploited by hackers. Testers execute the test cases one by one, sometimes using bug-tracking tools like Jira.

**5.   Report bugs**

When bugs are identified, the testing team will pass the metrics to the development team in the form of a[test report](https://www.globalapptesting.com/platform/test-results-analysis). This contains details on how many defects or bugs were found, how many test cases failed, and which need to be re-run.

**6.   Test again**

Once the development team has fixed the bugs, the software is handed back to the testers. They carry out the test cases again to check that the problem is resolved.