Test Technologies – Software testing

**What is software testing –**

software testing is an important process within the lifecycle of software development. It involves verifying and validating that a software application is free of bugs and meets the technical requirements set forth by its design and development. The process of software development is not only used to find faults within a program but is also used to measure and improve the software in terms of efficiency, accuracy and usability.

**What are the dangers of release untested software –**

There is a list of problems that can come from releasing untested software such as –

1. Poor quality – If the software isn’t properly tested or not tested enough it can result in unexpected errors and bugs, if the software isn’t a quality product the company could lose revenue from unhappy customers.
2. Security – With very strict rules regarding the safeguard of user information the software should be thoroughly tested for vulnerabilities and it safe from malicious attacks.
3. Poor software performance – If the website is not properly optimized for good performance the users could be very unforgiving when having to wait for site to load new information such as images and the response from the server, when the appropriate testing hasn’t been used it can result in user dissatisfaction and decreased productivity

**Types of software testing –**

Functional Testing –

Functional testing checks an application, website or system to ensure that it is doing exactly what it is meant to do. Ensuring that the software performs its intended functions correctly and meets the needs of the users. This could be implemented at the start of a project by creating a document listing functional or requirement specifications, essentially a list of what the website/software is supposed to do from the users perspective. The goal of functional testing is to validate the systems features, capabilities and interactions with different components, this would involve testing the software input and output, data manipulation, user interactions and the systems response to various scenarios and conditions.

Examples –

1. Unit Testing – This is preformed by writing scripts that test if individual components/units of an application match the requirements, this normally involves writing tests that call the methods in each unit and validate them when they return values that match the requirements.
2. Regression Testing – This test involves writing a test for a known bug and re-running this test after every change to the code base, this aims to immediately identify any change that reintroduces a bug.
3. Integration Testing - This would be used to validate that different software components, subsystems or applications work together as a system to achieve the desired functionality and performance. Integration testing helps to identify and resolve any issues that may arise when components are combined, such as compatibility issues, performance problems, incorrect communication or data corruption

Non-Functional Testing –

Non-Functional testing is a type of testing used to evaluate a software applications performance, usability, dependability and other non-functional characteristics. It is intended to test a systems readiness according to non-functional criteria that functional testing never considers. This software testing method enables quality assurance teams to check if the system complies with user requirements, increasing the products usability, effectiveness, maintainability.

Examples –

1. Performance testing – Eliminates the causes of the software’s sluggish and constrained performance. The software’s reading speed should be is quick as possible. There should be a detailed specification for the desired speed when using performance testing otherwise it won’t be evident if the test has been a success or a failure.
2. Load testing – The systems loading capability is tested during load testing. The systems can handle increasing simultaneous users because of its loading capacity.
3. Visual Testing – checks if every user sees the software user interface correctly, visual test verifies that each element on a web page has a proper shape, size and placement. Assesses the applications visible output and compares it to the outcomes anticipated by design, in other words, it assists in identifying visual bugs.