Introduction to Software Testing and Validation

Course Glossary

Throughout the course, you might find some new words, phrases and abbreviations being used – you can always come to this document to look up any unfamiliar terms!

Word	**Definition**
Acceptance testing	The final phase of testing, conducted by the end-users or clients, to ensure the software meets their requirements and functions as expected.
Alpha testing	A type of testing performed by the development team or internal testers to identify and fix bugs before the software is released to external testers or end-users.
Beta testing	A type of testing performed by a selected group of end-users to identify and fix bugs before the software is released to the general public.
Black-box testing	A testing method that focuses on the functionality of the software without considering its internal structure or implementation (how it's put together).
Boundary value analysis (BVA)	A test case design technique that focuses on testing the boundary values of input domains (range and/or areas) to identify defects.
Bug	An error, flaw, or unintended behaviour in software that causes it to produce incorrect or unexpected results.

Code coverage	A metric that measures the percentage of source code executed during testing, which assesses the thoroughness of testing efforts.
Debugging	The process of identifying, analysing, and fixing bugs in software.
Dynamic testing	A type of testing that involves executing the software to identify defects.
Equivalence partitioning (EP)	A test case design technique that divides input data into equivalent classes, allowing testers to reduce the number of test cases while maintaining test coverage.
Error	A human action or decision that produces incorrect or unexpected results.
Failure	The inability of software to perform its intended function or produce the expected results.
Functional testing	A type of testing that focuses on verifying that the software functions according to its requirements and specifications.
Gray-box testing	A testing method that combines elements of black-box and white-box testing, considering both the functionality and the internal structure of the software.

Integration testing	A testing phase that focuses on verifying that individual software components or modules work together correctly.
Load testing	A type of performance testing that assesses the software's ability to handle a specific number of tasks at the same time, i.e. end-users, transactions, or requests simultaneously.
Manual testing	The process of testing software by manually executing test cases and observing the results, without the use of automated tools.
Non-functional testing	A type of testing that focuses on assessing the software's performance, reliability, security, usability, and other non-functional aspects.
Performance testing	A type of testing that assesses the software's responsiveness, stability, and scalability under various conditions.
Quality assurance (QA)	A systematic approach to ensuring that a product or service meets specified requirements and is of consistent, high quality.
Regression testing	The process of retesting previously tested software components or features after changes have been made, to ensure that no new defects have been introduced.
Requirements	A set of documented specifications or conditions that a software product must meet or satisfy.

Security testing	A type of testing that focuses on identifying potential vulnerabilities, threats, and risks to the software and its data.
Smoke testing	A type of preliminary testing that verifies whether the software's basic functionality is working as expected.
Software development life cycle (SDLC)	A structured process that guides the development, testing, and deployment of software products.
Software testing	The process of evaluating a software product to identify defects, ensure quality, and verify that it meets specified requirements.
Static testing	A type of testing that involves examining the software without executing it, focusing on reviewing and analysing documentation, source code, and other artifacts.
Stress testing	A type of performance testing that evaluates the software's behaviour.
System testing	A type of testing that focusses on evaluating the software as a whole, including its integration with hardware, other software components, and external systems.
Test automation	The process of using specialised tools and software to execute test cases automatically, reducing the need for manual testing efforts.

Test case	A set of input values, execution conditions, and expected results used to verify whether a software application behaves as expected.
Test coverage	A metric that measures the extent to which the test cases cover different parts of the software, such as code, functionality, or requirements.
Test data	The set of input values and conditions used during the testing process to evaluate the software's behaviour and performance.
Test design techniques	Methods and strategies used to create test cases, such as boundary value analysis and equivalence partitioning.
Test environment	The hardware, software, and network configurations used to execute tests and evaluate software performance.
Test plan	A document that outlines the scope, objectives, and strategy for software testing, including test design techniques, test environments, and test schedules.
Test suite	A collection of test cases designed to validate specific aspects of the software.
Unit testing	A type of testing that focuses on evaluating individual software components or modules in isolation from the rest of the system.

Usability testing	A type of testing that focusses on assessing the software's user interface (UI) and overall user experience (UX).
Validation	The process of confirming that a software product meets its specified requirements and fulfils its intended purpose.
Verification	The process of checking that a software product has been developed according to its requirements, specifications, and design.
Walkthrough	A type of informal review process in which a developer or a team member presents the design, code, or any other software-related artifact to their peers, project stakeholders, or other interested parties.
White-box testing	A testing method that focusses on the internal structure, design, and implementation of the software, requiring knowledge of the source code.