General Discussion on Software Development Security for CISSP.

Software Testing.

Session 2 9/20/2020 Classroom-1

NOTES

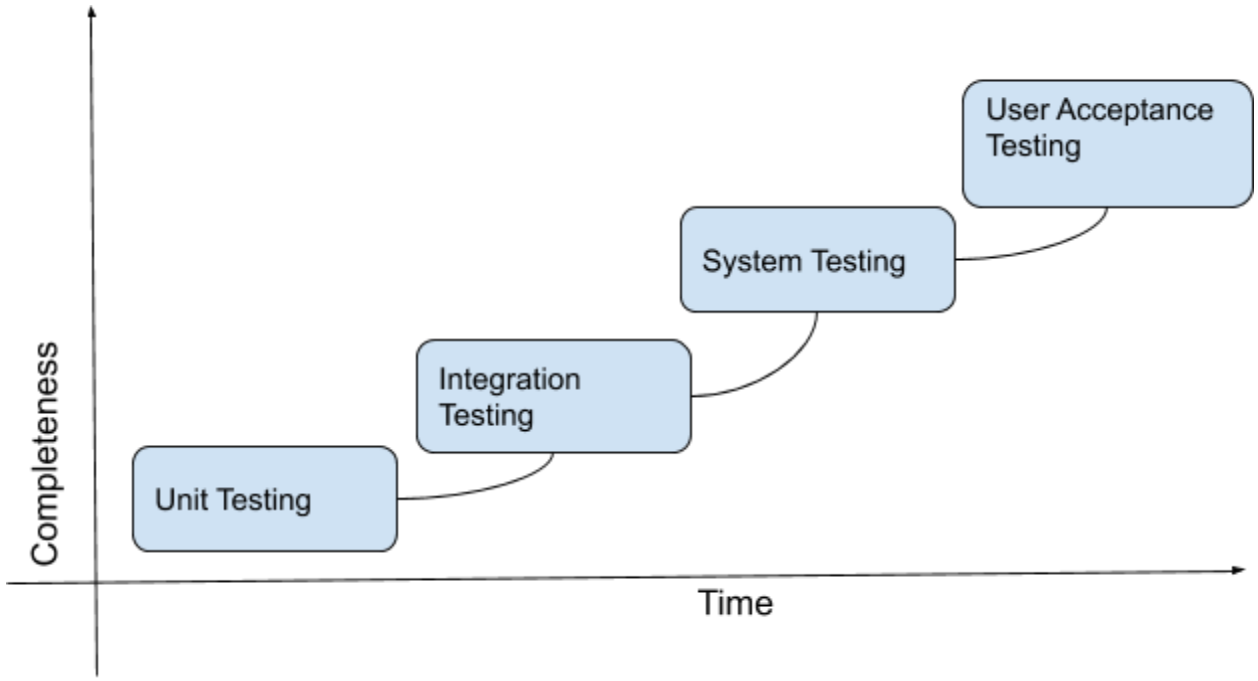
Durgesh Kalya CISSP

What is Software Testing?

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test

- Wikipedia

Lets see a more CISSP type definition of Software Testing:
In SDLC, we will come across Software testing in many different phases. It certainly starts after coding has started. There is a concept of Early Testing and is defined as initiation of testing in the early stages of SDLC in order to achieve more efficient, cost effective and high quality software. This usually starts at the Requirement Analysis phase.



Unit Testing:
UNIT TESTING, also known as COMPONENT TESTING, is a level of software testing where individual units / components of a software are tested. The purpose is to validate that each unit of the software performs as designed.

Integration Testing:
INTEGRATION TESTING is a level of software testing where individual units / components are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

System Testing:
SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

User Acceptance Testing:
ACCEPTANCE TESTING is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery (or writing that big check).

Requirements:

Functional Requirements:
Functional requirements define the basic system behaviour. Essentially, they are what the system does or must not do, and can be thought of in terms of how the system responds to inputs.
Examples: User Requirements, Features

Non-Functional Requirements:
Non-functional requirements specify how the system should do it. Non-functional requirements do not affect the basic functionality of the system (hence the name, non-functional requirements). Even if the non-functional requirements are not met, the system will still perform its basic purpose.
Example: User Experience, Overall Security

What is SRS?:
A software requirements specification is a description of a software system to be developed. It is modeled after business requirements specification, also known as a stakeholder requirements specification.

- It is a Document
- Describes what the software will do
- Describes how it will perform
- Describe functionality and needs to fulfill all stakeholders

What is RTM?:
A Requirement Traceability Matrix (sometimes called a Traceability Matrix or RTM) can be used to map customers' requirements to the software testing plan: it "traces" the "requirements", and ensures that they are being tested and followed.

The traceability matrix is typically a worksheet that contains the requirements with its all possible test scenarios and cases and their current state, i.e. if they have been passed or failed. This would help the testing team to understand the level of testing activities done for the specific product.

Test Coverage Analysis
It is a technique to ensure that your tests are testing your code or how much of your code you exercised by running the test.

Test Coverage Analysis allows for an estimation of the degree of testing conducted against a system or software to be calculated

Formula:

Test Coverage = Number of use cases tested / Total Number of use cases

Example:
If the total lines of code in a system component is 1000 and the number of lines being actually executed through all existing test cases is 650, then your test coverage is:
(650 / 1000) * 100 = 65%

Testing Methods:

Software Code Review and Testing

Black-box Testing:
Internal system design is not considered in this type of testing. Tests are based on the requirements and functionality. Input and output is observed.

White-box Testing:
Internal System design is considered for this type of testing. Tests are based on the knowledge about the internal logic of an application's code.

Grey-box Testing:
Gray Box Testing is a software testing method, which is a combination of both White Box Testing and Black Box Testing method.
It is a technique to test the software product or application with partial knowledge of the internal workings of an application

Dynamic Testing:
System is executed and its behavior is observed.

Static Testing:
The source code is observed without executing the system.

Manual Testing:
Is performed by humans. Humans guide every test and it is a manual process.

Automated Testing:
Test is performed by Applications.

Unit testing:
Testing of an individual software component or module is termed as Unit Testing. It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. It may also require developing test driver modules or test harnesses.

Integration testing:
Testing of a completed software system. A quality assurance team tests for components are working together as outlined in the SRS and design specifications.

Regression testing:
A specialist tests the software after changes are made. As exhaustive testing can be difficult, automated testing methods are used.

Software Penetration Testing:
Software penetration testing is done to check how the software or application or website is secure from internal and external threats. This testing includes how much software is secure from the malicious program, viruses and how secure and strong the authorization and authentication processes are. It also checks how software behaves for any hackers attack and malicious programs and how software is maintained for data security after such a hacker attack.

Blackbox, Whitebox and gray box testing is used by independent parties.

Acceptance Testing:
An Acceptance Test is performed by the client and verifies whether the system meets the end user requirements as per the business requirements and if it is as per the needs of the end-user. Client accepts the software only when all the features and functionalities work as expected.
It is the last phase of the testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

References:

Types of Testing:
https://www.softwaretestinghelp.com/types-of-software-testing/

Questions used for this session come from our Friend, please check out his website for all CISSP resources and more.
https://thorteaches.com/

Types of Testing:
https://softwaretestingfundamentals.com/

Integration Vs Interface testing:
https://www.tutorialspoint.com/differences-between-interface-and-integration-testing#:~:text=As%20mentioned%20in%20above%20point,interface%20to%20verify%20its%20functionality.&text=Integration%20testing%20due%20to%20its,manual%20as%20well%20as%20automated.

Combination Testing:
http://www.testingeducation.org/k04/ComboExamples.htm
Example: http://www.testingeducation.org/k04/examples/comb04s.html