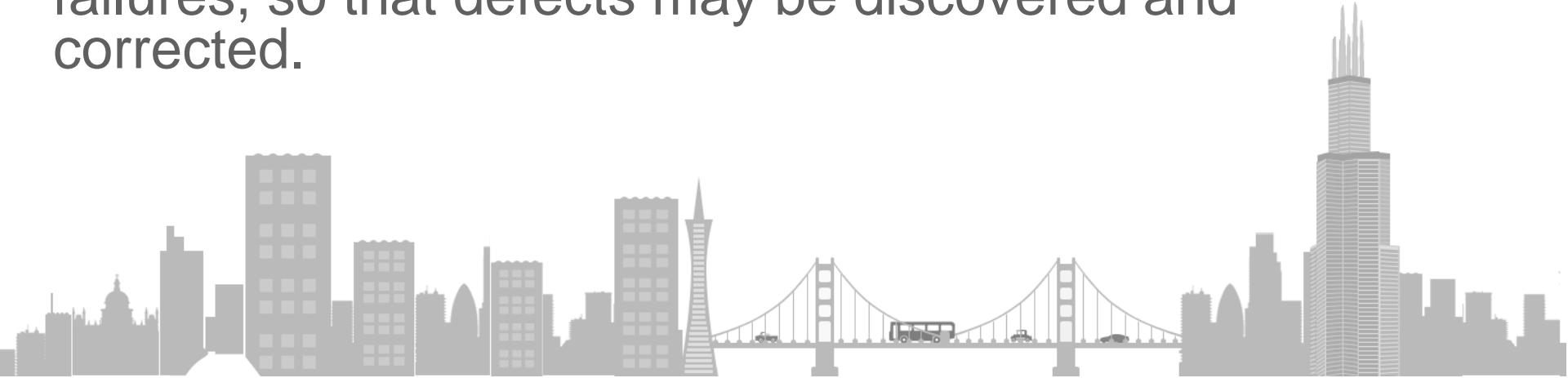


# Software Testing Concepts and Definition

Presented by Naseer Ahmad / Prepared by Sikandar Ahmed

Primary purpose of testing is to detect software failures, so that defects may be discovered and corrected.



## Items to be covered

- Concepts (Part 1)
  - SQA vs Testing
  - Test Senario vs Test Cases
  - Testing Cycle
- Testing Definitions (Part 2)
  - Testing Methods
  - Testing Levels
  - Testing Types

# Software Quality

---

- Quality software is reasonably bug or defect free, delivered on time and within budget, meets requirements and/or expectations, and is maintainable.
- ISO 8402-1986 standard defines quality as “the totality of features and characteristics of a product or service that bears its ability to satisfy stated or implied needs.”
- Key aspects of quality for the customer include:
  - Good design – looks and style
  - Good functionality – it does the job well
  - Reliable – acceptable level of breakdowns or failure
  - Consistency
  - Durable – lasts as long as it should
  - Good after sales service
  - Value for money

# Software Testing

---

- Software testing is a process of executing a program or application with the intent of finding the software bugs.
- It can also be stated as the process of validating and verifying that a software program or application or product:
  - Meets the business and technical requirements
  - Works as expected
  - Can be implemented with the same characteristic.

# Software Verification

---

- It makes sure that the product is designed to deliver all functionality to the customer.
- Verification is done at the starting of the development process. It includes reviews and meetings, walkthroughs, inspection, etc. to evaluate documents, plans, code, requirements and specifications.
- It answers the questions like: Am I building the product right? Am I accessing the data right (in the right place; in the right way).
- According to the Capability Maturity Model(CMMI-SW v1.1) we can also define verification as the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610].

# Software Validation

---

- Determining if the system complies with the requirements and performs functions for which it is intended and meets the organization's goals and user needs.
- Validation is done at the end of the development process and takes place after verifications are completed.
- It answers the question like: Am I building the right product? Am I accessing the right data (in terms of the data required to satisfy the requirement).
- Performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment.
- Determination of correctness of the final software product by a development project with respect to the user needs and requirements.
- According to the Capability Maturity Model(CMMI-SW v1.1) we can also define validation as The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. [IEEE-STD-610].

# Basic Testing Cycle

---

- Requirements Analysis
- Test Planning
- Test Development
- Test Execution
- Test Reporting
- Test Result Analysis
- Defect Retesting
- Regression Testing
- Test Closure





---

# Testing Methods

There are several approaches / techniques of Software  
Testing

# Testing Methods

---

- Static
- Dynamic
- Black Box
- White Box
- Visual (GUI)

# Testing Methods (Cont.)

---

## Static Testing:

- Static testing is the testing of the software work products manually, or with a set of tools, but they are not executed.
- It starts early in the Life cycle and so it is done during the verification process.
- It does not need computer as the testing of program is done without executing the program. For example: reviewing, walk through, inspection, etc.

# Testing Methods (Cont.)

---

## Dynamic Testing:

- Dynamic testing is the testing of the dynamic behaviour of code.
- It involves working with the software, giving input values and checking if the output is as expected by executing specific test cases which can be done manually or with the use of an automated process.
- It is done during Validation process.
- The software is tested by executing it on computer. Ex: Unit testing, integration testing, system testing.

# Testing Methods (Cont.)

---

## Black Box Testing:

- Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation.
- The tester is only aware of what the software is supposed to do, not how it does it.
- It also known as Specification-based testing technique or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.

# Testing Methods (Cont.)

---

## White Box Testing:

- White-box testing tests internal structures or workings of a program
- It is also known as Structure-based or 'glass-box' testing technique because here the testers require knowledge of how the software is implemented, how it works.
- In white-box testing the tester is concentrating on how the software does it.

# Testing Methods (Cont.)

---

## Visual (GUI) Testing:

- GUI testing is the process of testing a product's graphical user interface to ensure it meets its written specifications like testing images and buttons alignment on any webpage.



---

# Testing Levels

Each phase of SDLC goes through the testing. Hence there are various levels of testing.



# Testing Levels

---

- Unit Testing
- Component Testing
- Integration Testing
- System Testing
- Acceptance Testing
- Alpha Testing
- Beta Testing

# Testing Levels (Cont.)

---

## Unit Testing:

- Unit testing is a method by which individual units of source code together with associated control data are tested to determine if they are fit for use.
- A unit is the smallest testable part of an application like functions/procedures, classes, interfaces.
- Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

# Testing Levels (Cont.)

---

## **Component Testing:**

- Component testing is also known as module and program testing.
- Component testing is the phase in which individual software modules are tested to find defects in the module and verifies the functioning of software Module.
- Component testing may be done in isolation from rest of the system depending on the development life cycle model chosen for that particular application.

# Testing Levels (Cont.)

---

## Integration Testing:

- Integration testing is the phase in which individual software modules are combined and tested as a group to make verify integrated system is ready for system testing.
- It occurs after unit testing and before validation testing.
- Integration testing is done by a specific integration tester or test team.

# Testing Levels (Cont.)

---

## System Testing:

- System testing is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
- System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.
- System testing is most often the final test to verify that the system to be delivered meets the specification and its purpose.
- System testing should investigate both functional and non-functional requirements of the testing.

# Testing Levels (Cont.)

---

## Acceptance Testing:

- After the system test has corrected all or most defects, the system will be delivered to the user or customer for acceptance testing.
- Acceptance testing is a test conducted to determine if the requirements of a specification or contract are met prior to its delivery.
- Acceptance testing is basically done by the user or customer although other stakeholders may be involved as well.

# Testing Levels (Cont.)

---

## Alpha Testing:

- Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site.
- Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.
- This test takes place at the developer's site.

# Testing Levels (Cont.)

---

## **Beta Testing:**

- It is also known as field testing. It takes place at customer's site.
- It sends the system to users who install it and use it under real-world working conditions.
- A beta test is the second phase of software testing in which a sampling of the intended audience tries the product out.
- Versions of the software, known as beta versions, are released to a limited audience outside of the programming team.





---

# Testing Types

A test type is focused on a particular test objective

# Testing Types (Cont.)

---

## **Functional Testing:**

Functional testing refers to activities that verify a specific action or function of the code.

Functional tests tend to answer the question of "can the user do this" or "does this particular feature work."

## **Non Functional Testing:**

Non-functional testing refers to aspects of the software that may not be related to a specific function or user action, such as scalability or other performance, behavior under certain constraints, or security.

# Testing Types (Cont.)

---

## Functional Testing

- Installation
- Development
- Usability
- Sanity
- Smoke
- Regression
- Destructive
- Recovery
- Automated
- User Acceptance

## Non Functional

- Compatibility
- Performance
- Security
- Accessibility
- Internationalization / Localization

# Testing Types (Cont.)

---

## **Installation Testing:**

- Installation testing focuses on what customers will need to do to install and set up the new software successfully.
- The testing process may involve full, partial or upgrades install/uninstall processes.
- This testing is typically done by the software test engineer in conjunction with the configuration manager.

# Testing Types (Cont.)

---

## **Development Testing:**

- Development Testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs.
- It is performed by the software developer or engineer during the construction phase of the software development lifecycle.
- Development Testing might include static code analysis, data flow analysis metrics analysis, peer code reviews, unit testing, code coverage analysis, traceability, and other software verification practices.

# Testing Types (Cont.)

---

## **Usability Testing:**

- Usability testing is a technique used in user-centered interaction design to evaluate a product by testing it on users.
- This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system.
- Usability testing generally involves measuring how well test subjects respond in four areas: efficiency, accuracy, recall, and emotional response.

# Testing Types (Cont.)

---

## **Sanity Testing:**

- The sanity test which offers quick, broad, and shallow testing determines whether it is possible and reasonable to proceed with further testing.
- If the sanity test fails, it is not reasonable to attempt more rigorous testing.
- Sanity tests are ways to avoid wasting time and effort by quickly determining whether an application is too flawed to merit any rigorous testing.

# Testing Types (Cont.)

---

## **Smoke Testing:**

- Smoke testing is preliminary testing to reveal simple failures severe enough to reject a prospective software release e.g. smoke test may ask basic questions like "Does the program run?", "Does it open a window?"
- The purpose is to determine whether the application is so badly broken that further testing is unnecessary.
- Smoke testing performed on a particular build is also known as a build verification test.
- Smoke testing is done by developers and testers both.
- A smoke test is used as an acceptance test prior to introducing a new build to the main testing process.



# Testing Types (Cont.)

---

## **Regression Testing:**

- Regression testing is a type of software testing that seeks to uncover new software bugs, or regressions, in existing areas of a system after changes (such as enhancements, patches or configuration changes) have been made to them.
- Common methods of regression testing include rerunning previously completed tests and checking whether program behavior has changed and whether previously fixed faults have re-emerged.

# Testing Types (Cont.)

---

## **Destructive Testing:**

- Destructive software testing which attempts to cause a piece of software to fail in an uncontrolled manner, in order to test its robustness.
- It verifies that the software functions properly even when it receives invalid or unexpected inputs, thereby establishing the robustness of input validation and error-management routines.

# Testing Types (Cont.)

---

## **Recovery Testing:**

- Recovery testing is the activity of testing how well an application is able to recover from crashes, hardware failures and other similar problems.
- E.g. While an application is receiving data from a network, unplug the connecting cable. After some time, plug the cable back in and analyze the application's ability to continue receiving data from the point at which the network connection disappeared.

# Testing Types (Cont.)

---

## **Automated Testing:**

- Test automation is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes to predicted outcomes.
- Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to perform manually.

# Testing Types (Cont.)

---

## **User Acceptance Testing:**

- User acceptance testing (UAT) consists of a process of verifying that a solution works for the user.
- It is not system testing, but rather is there to ensure that the solution will work for the user i.e. test the **user accepts** the solution.

# Testing Types (Cont.)

---

## **Compatibility Testing:**

- Compatibility testing, part of software non-functional tests, is testing conducted on the application to evaluate the application's compatibility with the computing environment.
- Computing environment may contain different OS types (IOS 6, IOS 7), different types of browsers (Chrome, Firefox, IE).

# Testing Types (Cont.)

---

## Performance Testing:

- Performance testing is generally executed to determine how a system or sub-system performs in terms of responsiveness and stability under a particular workload.
- It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.
  - **Load Testing** is a testing that the system can continue to operate under a specific load, whether that be large quantities of data or a large number of users. This is generally referred to as software scalability.
  - **Volume testing** is a way to test software functions even when certain components (for example a file or database) increase radically in size.
  - **Stress testing** is a testing beyond normal operational capacity, often to a breaking point, in order to observe the results. It is a form of software testing that is used to determine the stability of a given system.

# Testing Types (Cont.)

---

## **Security Testing:**

- Security testing is a process to determine that an information system protects data and maintains functionality as intended.
- The six basic security concepts that need to be covered by security testing are:
  - Confidentiality
  - Integrity
  - Authentication
  - Availability
  - Authorization
  - non-repudiation.



# Testing Types (Cont.)

---

## **Accessibility Testing:**

- Accessibility is the degree to which a product, device, service, or environment is available to as many people as possible.
- This testing may include compliance with standards such as World Wide Web Consortium (W3C).
- Accessibility is not to be confused with usability

# Testing Types (Cont.)

---

## **Internationalization and localization Testing:**

- Internationalization and localization are means of adapting computer applications to different languages, regional differences and technical requirements of a target market.
- Internationalization is the process of designing a software application so that it can be adapted to various languages and regions without engineering changes.
- Localization is the process of adapting internationalized software for a specific region or language by adding locale-specific components and translating text.

# References

---

**Institute of Software Testing Qualifications Board**

<http://istqbexamcertification.com/>

**ISO/IEEE Standards**

<http://standards.ieee.org/findstds/standard/29119-1-2013.html>

<http://www.softwaretestingstandard.org/>

**QAI Global Institute**

<http://www.qaiusa.com/>

**Wikipedia:**

[http://en.wikipedia.org/wiki/Software\\_testing](http://en.wikipedia.org/wiki/Software_testing)



---

# Thank You