Overview

What is Software Testing ?

Definition

Software testing is a process used to identify the correctness, completeness and quality of developed computer software. It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to end users.

What is Software Testing ?

Definition

In simple words, software testing is an activity to check that the software system is defect free

What is Software Testing ?

Definition

Testing is executing a program on a tiny sample of the input domain

Two Important aspect of testing:

1. Dynamic technique
2. Optimistic approximation

Why is Testing Important ?

Airbus A300

264 People dead

Why is Testing Important ?

THERAC -25 RADIATION THERAPY

3 Dead and 3 Critically Injured

Why is Testing Important ?

Failed Satellite Launch

1.2 billion lost

Why is Testing Important ?

- Software is buggy!
  - Cost of bugs: 60 B/year US Economy
  - On average: 1-5 bugs/KLOC
  - 100 % correct mass market software is impossible

Why is Testing Important ?

As you can see testing is important because software bugs can be expensive or even dangerous

Causes of Software Defects

- Faulty requirements definition
- Time Pressure
- Complex Code
- Many System Interactions
- Coding errors
- Complexity of Infrastructure
- Changing technologies
- Non compliance with standards

Role of Testing in Software Development

- Testing of systems and documentation can help to reduce the risk of problems occurring during operation and contribute to the quality of the software system
- Software testing may also be required to meet contractual or legal requirements, or industry specific standards

Testing and Quality

- Testing ensures that key functional and non functional requirements are met
- Testing measures the quality of software in terms of the number of defects found, the tests run, and the system covered by the tests
- Do you think testing increases the quality of the software ?

Testing and Quality

- Testing cannot directly enhance quality. Testing can give confidence in the quality of the software if it finds fewer or no defects
- Understanding the root causes of defect found
- Testing is one component in the overall quality assurance activity.

Tester Roles

Quality Assurance

Quality Assurance(QA) is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers

Tester Roles

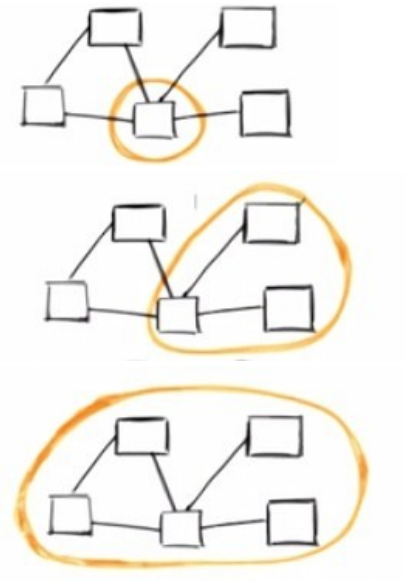
- Functional Tester-is responsible for checking if the product works based on business requirements
- Automation Tester-is responsible for writing script that automate testing rather than manual testing the functionality
- Performance Tester-is responsible for testing the product on different load and report the finding the developer
- Mobile Tester- is responsible for testing the product on different mobile devices
- Pen Tester-is responsible for exposing all security flaws(SQL Injection, Cross scripting, session hijacking,etc) in the application

Testing Levels

Unit Testing

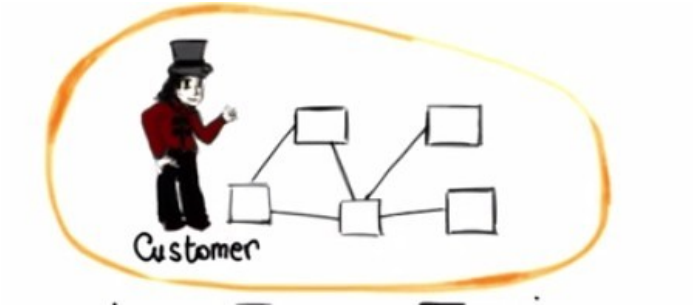
Integration Testing

System Testing



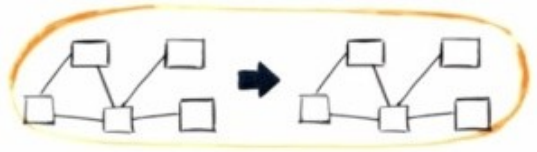
Testing Levels

Acceptance Testing



Validation Testing

- Regression Testing
- Alpha Testing
- Beta Testing



Unit Testing

- Algorithms and logic
- Data structures(global and local)
- Interfaces
- Independent paths
- Boundary conditions
- Error handling

Integration Testing

Why Integration Testing ?

- One module can have an adverse effect on another
- Sub functions, when combined, may not produce the desired major function
- Individually acceptable imprecision in calculation may be magnified to unacceptable levels
- Interfacing errors not detected in unit testing may appear
- Timing problems(in real-time systems) are not detectable by unit testing
- Resource contention problems are not detectable by unit testing

Validation Testing

- Determine if the software meets all of the requirement defined in the SRS
- Having written requirements is essential
- Regression testing is performed to determine if the software still meets all of its requirements in light of changes and modifications to the software
- Regression testing involves selectively repeating existing validation tests, not developing new tests

Alpha and Beta Testing

- Its best to provide customers with an outline of the things that you would like them to focus on and specific test scenarios for them to execute.
- Provide with customers who are actively involved with a commitment to fix defects that they discover

Acceptance Testing

- Similar to validation testing except that customers are present or directly involved.
- Usually the tests are developed by the customer

Test Types

A test type is focused on a particular test objective, which could be any of the following:

- Functional Testing
- Non-Functional Testing
- Structural Testing
- Retesting related to Changes

Functional Testing

- The functions are what the system does
- Many be described in work products such as a requirements speciffication, use cases, or a functional specification
- Types of functional testing include:
  - Security Testing- Investigates the functions relating to detection of threats, such as viruses,from malicious outsiders
  - Interoperability Testing - Evaluates the capability of the software product to interact with one or more specified components or systems.

Non-Functional Testing

- It is the testing of ”how” the system works
- Non-functional testing includes, but is not limited to:
  - Performance Testing- How many users can connect to the system and how will that affect the performance of the software
  - Load Testing - How will the system perform if we do a single transaction so many times
  - Stress Testing - How will the system perform under very tough circumstances, many users, many transactions, low memory,..etc Usability Testing- Is the system easy to use
  - Maintainability Testing- Is the system easy to maintain if we need to fix a defect
  - Reliability Testing- Is the system reliable or does it crash eventually
  - Portability Testing - Is the system easy to port from one platform to another

Structural Testing

- Coverage is the exten that a structure has been exercised by a testing, expresssed as a percentage of the items being covered
- If coverage is not 100%, then more tests may be designed to test those items that were missed to increase coverage.

Retesting related to Changes

- After a defect is detected and fixed, the software should be re-tested to confirm that the original defect has been successfully removed. This called confirmation testing or retesting
- Regression testing is the repeated testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes

Testing Types Vs Test Levels

What kind of test types will be performed at each test levels ?

- Functional Testing
- Non-functional Testing
- Structural Testing
- Retesting related to changes

Testing Techniques

Two Important techniques of testing:

1. Black box testing
2. White box testing

Black Box Testing

- Based on a description of the software(specification)
- Cover as much specified behavior as possible
- can not reveal errors due to implementation details

Black Box Testing Limitation

Specification: inputs an integer and print it

Example:

```
1. void printNumBytes(param) {
2.              if(param < 1024) printf("%d",param)
3.              ;else printf("%dKB",param / 123);
4. }
```

Limitation

It doesn't reveal errors hidden in the implementation details

White Box Testing

- Based on the code
- Cover as much coded behavior as possible

White Box Testing Limitation

Specification: inputs an integer param and returns half of its value if even, its value otherwise

Example:

```
1. void fun(int param) {
2.          int result;
3. result=param/2;
4. return result;
5. }
```

Limitation

It doesn't reveal errors due to missing paths

Seven Testing Principles

1. Testing shows presence of defects
2. Exhaustive testing is impossible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence of errors fallacy

Stages of Testing

- Test Planning
- Analysis and Design Implementation
- and Execution Evaluating exit criteria
- and Reporting Test Closure activities
-

Test Planning Activities

- Defining the Scope of Project
- Determine Test Approach
- Defining Test Strategy
- Determine Test Resources
- Define Exit Criteria

Analysis and Design

- Analysis of Business requirements
- Designing Test Scenarios
- Creating Test Data
- Designing Testcases based on Scenarios
- Test Environment Setup

Analysis and Design

- Executing the Testcases in givenTimeline
- Prioritizing the Testcases
- Writing Automation scripts if necessary
- Defect Logging and Tracking
- Test Case Status Reporting

Evaluating exit criteria and Reporting

- To check the test Status against the exit criteria specified in test planning.
- To assess if more test are needed or if the exit criteria specified should be changed.
- To write a test summary report for stakeholders.

Evaluating exit criteria and Reporting

- To finalize and archive testware such as scripts, test environments, etc. for later reuse.
- To handover the testware to the maintenance organization. They will give support to the software.
- To evaluate how the testing went and learn lessons for future releases and projects.

Testing Methodologies

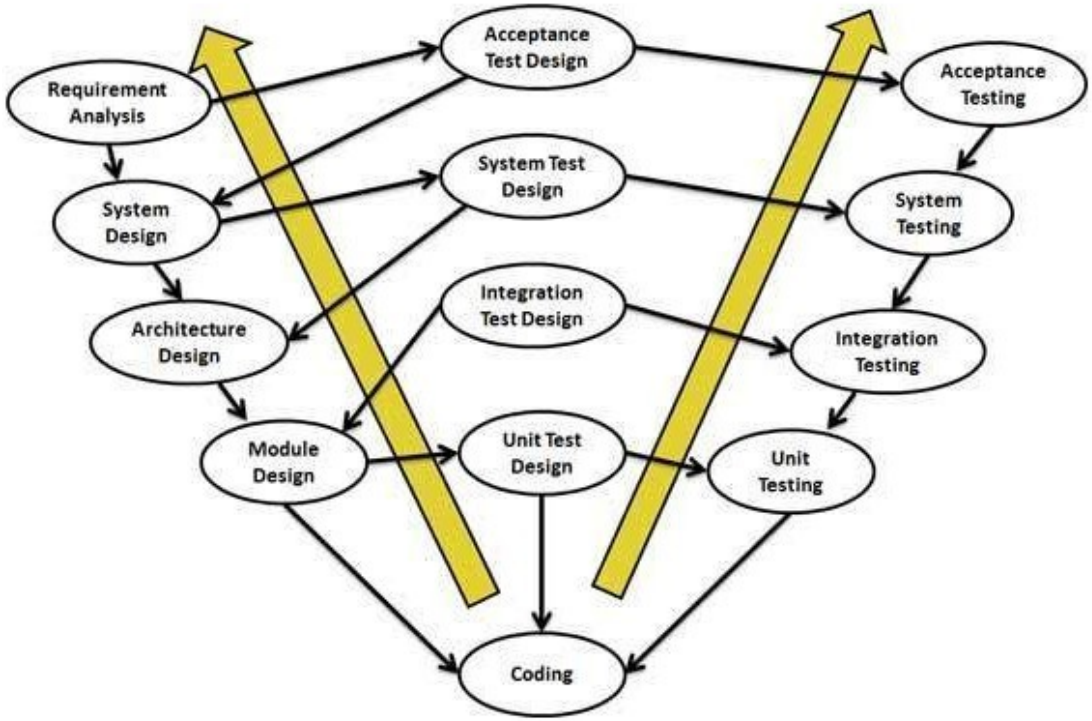
- Waterfall model
- Iterative Model
- Agile

Verification and validation VandV

check during development if the product:

- Meets its specifications
- Delivers the functionality expected by the people paying for this product

The classical V model



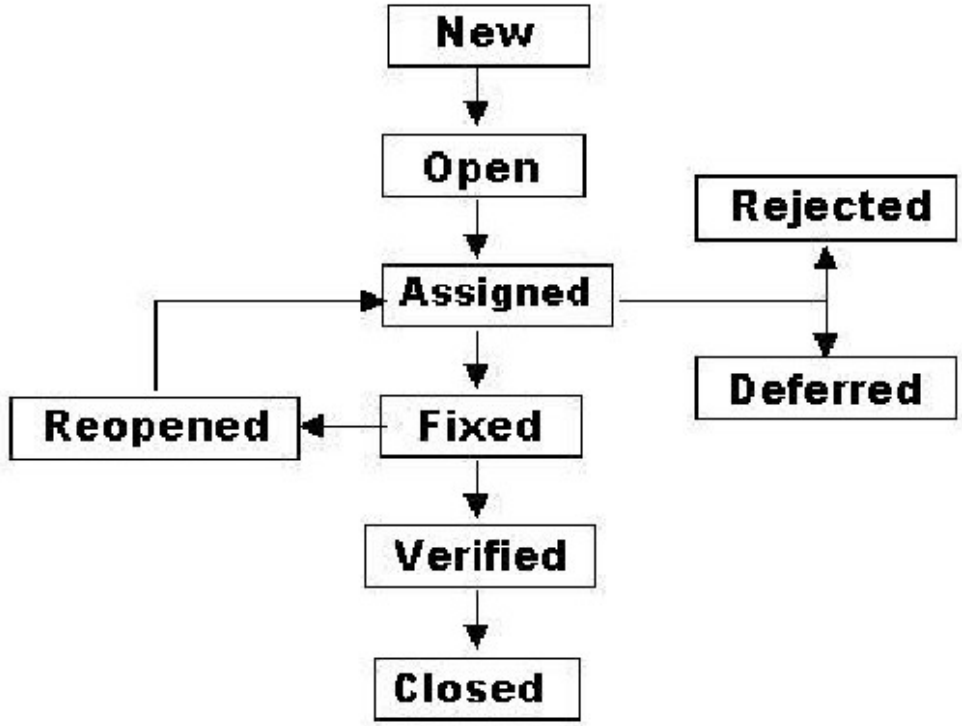
Verification

- evaluates a component or system to determine whether the products of a given development phase satisfy the conditions imposed at the start of each phase
- Are we building the system right?

Validation

- Process of evaluating a system or component during or at the end of development process to determine whether it satisfies specified requirements
- Are we building the right system?

Defect Life Cycle



Defect Life Cycle

- New: QA raised the Defect -
- Open : Test Manager/Developer changes the status
- Assigned – Developer – he will work on it
- Fixed- 26th February Build
- Verification : Once you verify
- Closed

Defect Template

Description:	
Steps to Reproduce:	
Test Data:	
Expected Result:	
Action Result:	
Screen shot:	
Time stamp:	
Assigned to:	
Severity :	
Priority :	

Defect Template Example

Description:	Password reset happening after clicking on cancel
Steps to Reproduce:	login till clicking on cancel
Test Data:	user id , password
Expected Result:	Password reset should not happen
Action Result:	Password reset is happening
Screen shot:	image or video
Time stamp:	
Assigned to:	Test Manager/Developer
Severity :	low medium high
Priority :	high