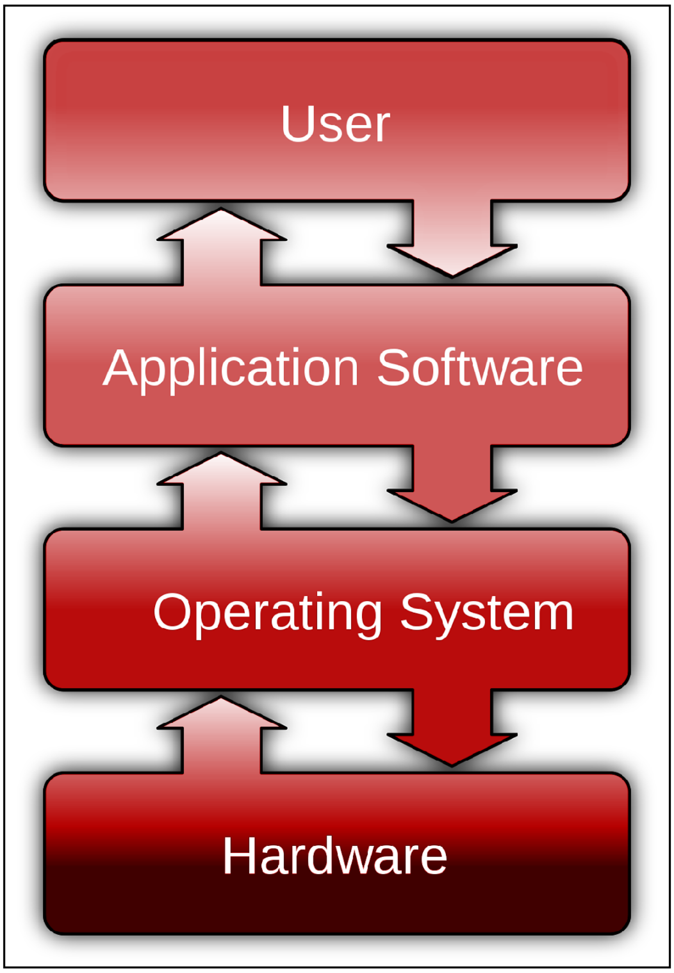
What is an application ?

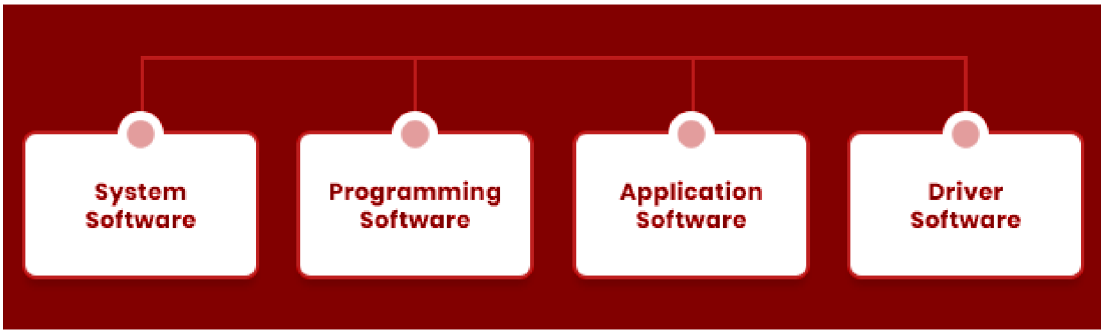
An Application is a combination of Hardware and Software to perform specific tasks.



What is software?

A Software is a collection of computer programs that helps us to perform a task.

Types of Software:



System software- OS, Servers.

Programming software- Compilers, Debuggers, interpreters.

Application software- Industrial automation, Business software, Games, Telecoms.

Software Architecture

One tier architecture

Two tier architecture

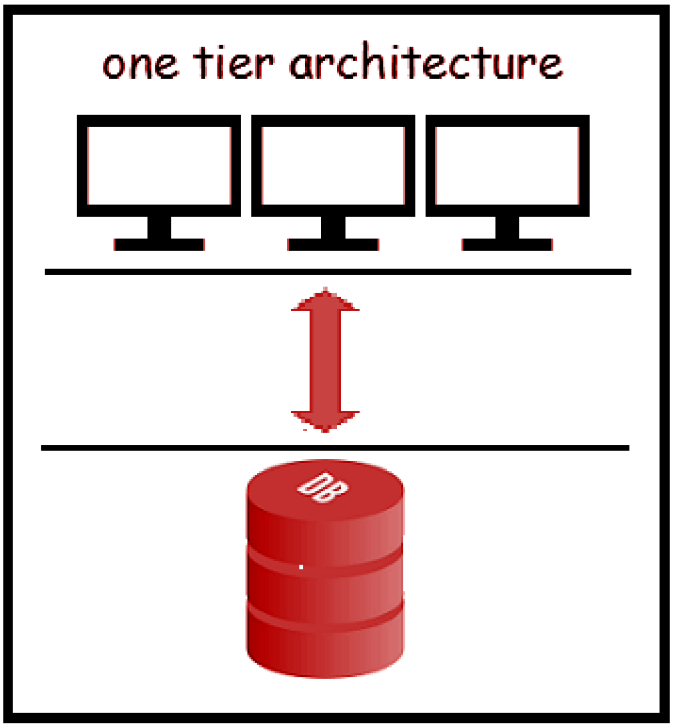
Multi tier or Three tier architecture

Examples:

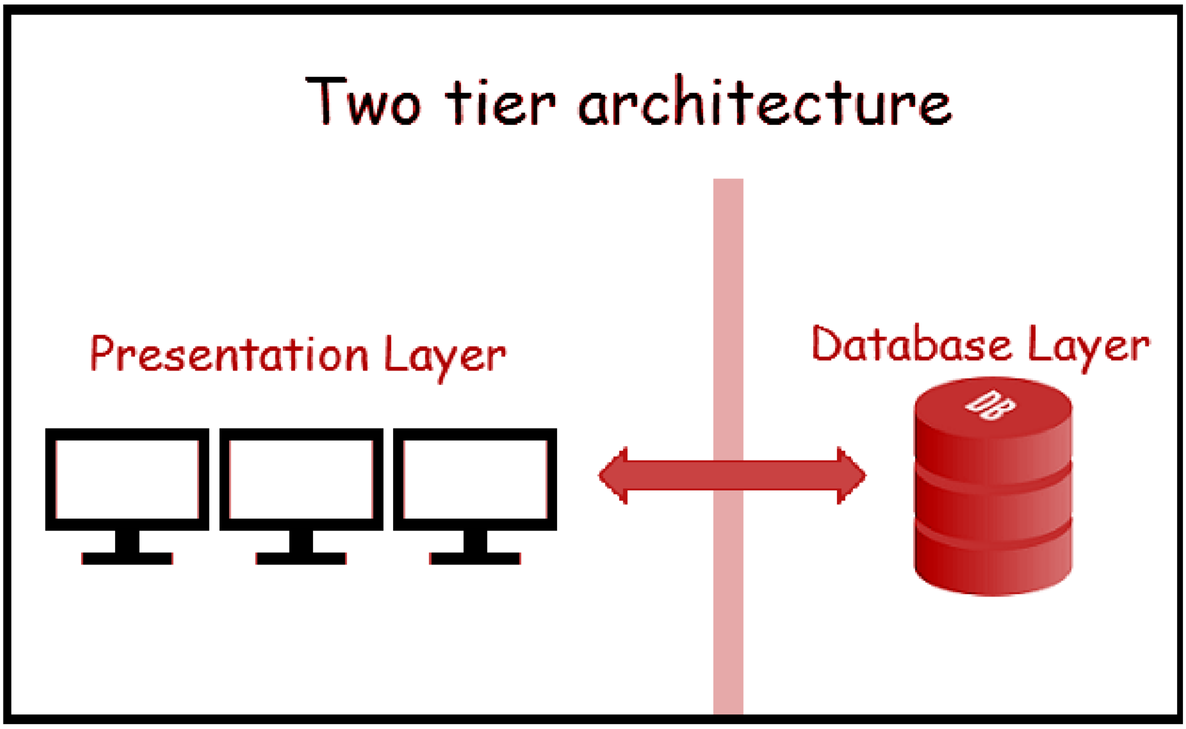
One tier architecture- ms-office, paint

Two tier architecture- Railway ticket counter

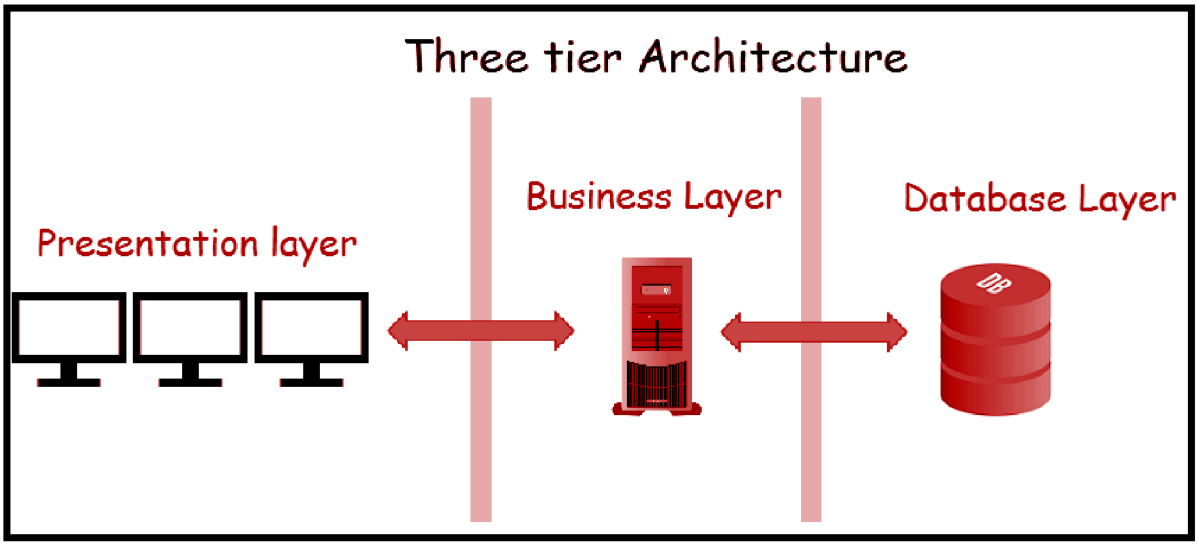
Multi tier architecture- Gpay



one tier architecture
DB



Two tier architecture
Presentation Layer
Database Layer
DB



Why do we need testing?

Ensure that software is bug free.

Ensure that the system meets customer requirements and software specifications.

Fixing the bugs identified after release is expensive.

What is Software Testing?

Software Testing is a part of the software development process.

Software Testing is an activity to identify the defects in the software.

The objective of testing is to release good quality products to the client.

What is Software Quality?

Bug Free
Delivered on time
Within Budget
Meet Requirements
Meet Expectation
Maintainable

Software company teams and roles?

Management Team -→ Business Analyst(BA), Software Architect(SA), Project Manager(PM)

Designing Team → Team lead, designers,

Developing Team → Developers

Testing Team → Team lead, testers

Operations Team or Supporting team

Testing Environment

Dev ------- Unit and Integration Testing
QA --------- System Testing
Staging--- UAT
Production---- Live

Product Vs Project

If a software application is developed for specific customer requirements then it is called **Project**.

If a software application is developed for multiple customer requirements then it is called **Product**.

Testing Documents:

- SRS(Software Requirement Specifications)
- Test Plan
- Test case
- Defect Report
- RTM(Requirement Traceability matrix)
- Test Summary Report

SRS- SRS is a document that describes what the software will do and how it will be expected to perform. It is also describes the functionality of the product needs to fulfill all business needs

Test Plan- Test Plan is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product.

Test Case- Test case is a set of actions executed to verify a particular feature or functionality of a software application.

RTM- is a document that maps and traces test cases with requirements and defects. Two types of RTM

Forward RTM - Mapping test cases with Requirement

Backward RTM - Mapping test cases with Defect

Test Scenario- It is a high level of test case. It describes the what action going to done in test case.

Test Coverage- It measures how much your tests are covering things like test requirements,code,different user scenarios. It is a useful metric for measuring the effectiveness of your testing efforts.

What is build ?

Build refers to converting source code of a program to executable code that can be run on a computer.

What is deployment ?

Deployment is the mechanism through which applications, modules, updates, and patches are delivered from developers to testers from one environment to another.

What is release ?

It is distribution of final version of software application

Error, defect,bug & failure

Error: Any incorrect human action that produces a problem in the system is called an error.

Defect: Deviation from the expected behavior to the actual behavior of the system is called defect.

Bug: When Developer accepts a defect is valid then it is known as a bug.

Failure: Total system or application collapse during testing is called failure.

Diff B/w Priority and Severity

Priority- How quickly we want to fix that particular bug.(low,medium,high,very high)

Severity- How badly bugs affected the software's functionality.(low,medium,high,very high)

Static V/S Dynamic Testing

Static testing is an approach to test project documents in the form of Reviews,Walkthroughs and Inspections.It is a verification process.

Dynamic testing is an approach to test the actual software by giving inputs and observing results. It is a validation process.

Reviews, Walkthrough & Inspection

Reviews:

Conducts on documents to ensure correctness and completeness.

Requirement Reviews

Test plan reviews

Test cases reviews

Walkthroughs:

It is a formal review and we can discuss/raise the issues.It can happen at any time and conclude just like that with no schedule as such.

Inspections:

It's a formal approach to the requirements schedule. At least 3- 8 people will sit in the meeting. Inspection will have a proper schedule which will be intimated via email to the concerned developer/tester.

SDLC(Software Development Life Cycle)

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares.

The SDLC aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within time and cost estimates.

SDLC phases

Requirement Gathering

Planning

Designing

Developing

Testing

Deployment and Maintenance

STLC(Software Testing Life Cycle)

Software Testing Life Cycle (STLC) is a sequence of specific activities conducted during the testing process to ensure software quality goals are met. STLC involves both verification and validation activities.

STLC phases

Requirement Analysing
Test Plan Design
Test Case Design
Test Execution
Test Log
Defect Tracking Process
Test Closure

Explain about your testing life cycle ?

Once our team receives an SRS document my team lead analysis the requirement and prepares the Test plan and RTM.
As a tester I receive SRS and test plan. I will analyze the requirements and identify test scenarios.
Then I write test cases and review my test cases.Once I receive a build from the developing team I will execute my test cases.
If I identify any defect. I will perform a Defect tracking process.Once bugs are fixed i perform retest and close the bug and Update in RTM.
End of the project my Team lead prepared a Test summary report.

SRS Content:

Introduction

Scope

Objective

Environment

Functional Requirement

Non Functional Requirement

Use Case Diagram

Energy Flow Diagram

Summary

Test plan Content:

Scope

Strategy

Environment

Control(Review Meeting,Change Request Meeting,Defect Review Meeting)

Function to be tested

Resources and responsibilities

Deliverables(Bug report,updating RTM, Test summary Report)

Entry Criteria & Exit Criteria

Suspension Criteria & Resumption Criteria

Summary

Test Case Content:

Test Scenario
Test Scenario ID
Test case ID
Priority
Condition
Action
Test Data
Expected Result
Actual Result
Status

Defect Report or Bug Report:

Bug ID

Test Scenario ID

Test Case ID

Bug Description

Priority

Severity

Build Version

Release Version

Status

RTM Content:

Module

Sub Module

Requirement ID

Test Strategy

Test Scenario

Test Scenario ID

Test Case ID

Defect ID

Status

Comments

Release Date

Testing Principles:

- Testing shows the presence of defects
- Exhaustive Testing is not possible
- Early Testing
- Defect Clustering
- Pesticide Paradox
- Testing is context-dependent
- Absence of errors fallacy

Testing shows the presence of defects

Testing talks about the presence of defects and doesn't talk about the absence of defects. Even if no defects are found in an application it is not a proof of correctness.

Exhaustive Testing is not possible

Sometimes it's very hard to test all the modules and their features effectively. Instead, we need the optimal amount of testing based on the risk assessment of the application.

Early Testing

Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in the early stages of testing.

Defect Clustering

Defect Clustering which states that a small number of modules contain most of the defects. approximately 80% of the problems are found in 20% of the modules.

Pesticide Paradox

If we are executing the same set of test cases again and again over a particular time, then these kinds of the test will not be able to find the new bugs in the software or the application.it is very important to review all the test cases frequently.

Testing is context-dependent

This principle states that we have multiple fields such as e-commerce websites, commercial websites are available in the market.To check this type of application, we will take the help of various kinds of testing, different techniques, approaches, and multiple methods. Therefore, the testing depends on the context of the application.

Absence of errors fallacy

An application is completely tested and there are no bugs identified so that the application is 99 percent bug-free.But there is the chance application is tested with incorrect requirements,The absence of error fallacy means identifying and fixing the bugs would not help if the application is impractical and not able to accomplish the client's requirements and needs.

Levels of Software Testing

Unit Testing
Integration Testing
System Testing
User Acceptance Testing(UAT)

Testing Techniques

White box Testing
Black box Testing
Grey box Testing

What is white box testing?

Testing the application with knowledge of programming skills. Unit testing, integration testing done by developer.

What is Black box Testing?

Testing the application without knowledge of coding or programming skills.

To check functional behavior of the application. It’s perform system testing and UAT testing done by tester

Grey box Testing:

Combination of white box and black box testing techniques.

Unit Testing:

It is white box testing technique.
It is conducted by the developers.
A unit is the smallest testable part of software.
It is conducted on a single program or single module.

Unit testing techniques:

- Basis path testing
- Control structure testing
- Conditional coverage
- Loops Coverage

Integration Testing:

It is white box testing technique.
It is conducted by both developers and testers.

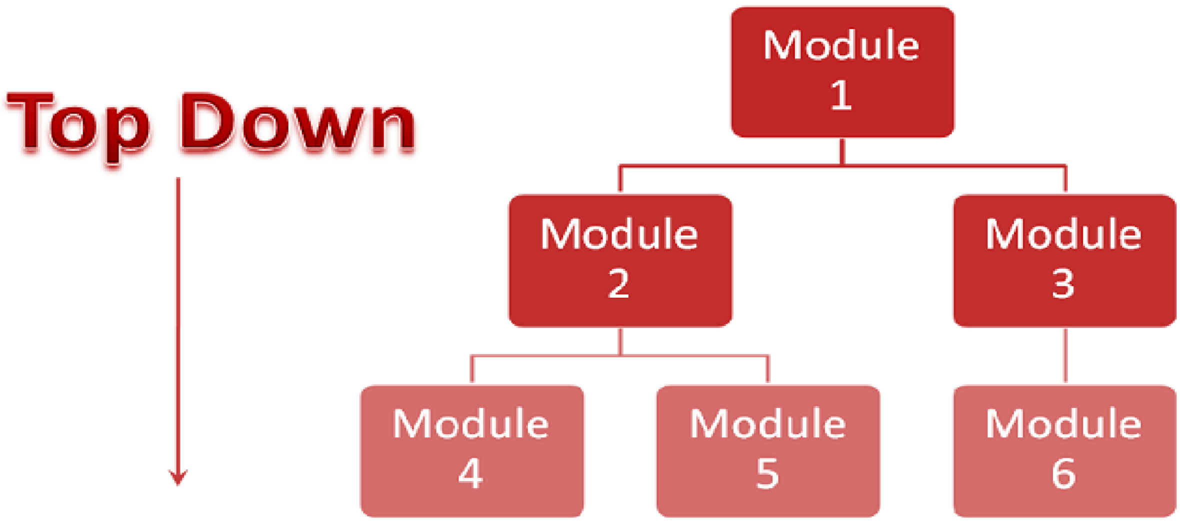
Integration testing focuses on communication Flow and Data flow between two different modules.

Approaches:

Top Down Approach
Bottom Up Approach
Sandwich Approach
Bigbang

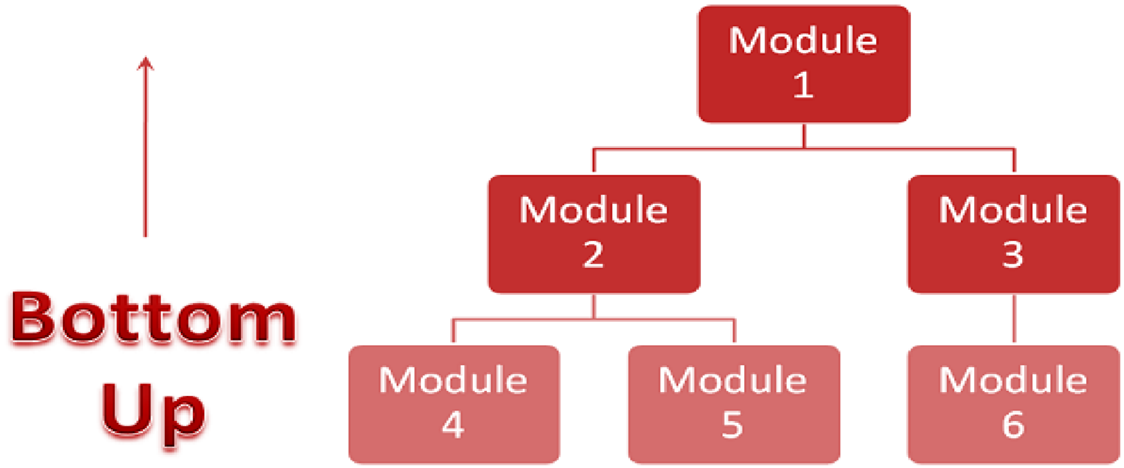
Top Down Approach

In the Top to down approach, testing takes place from top to down following the control flow of the software system.



Bottom Up Approach

In the Top to down approach, testing takes place from down to up following the control flow of the software system.



Sandwich Approach

Combination of Topdown and bottom up is called the sandwich approach.

BigBang

Dummy modules created to validate those modules with the help of stubs and drivers.

System Testing:

It is a black box testing technique.

This testing is conducted by a testing team.

It validates the complete flow of an application. The purpose of a system testing is to evaluate the end-to-end system specifications.

It is testing the overall behaviour of the application in depth.

System Testing focuses on below aspects.

- Graphical User Interface Testing (GUI)
- Functional Testing
- Non-Functional Testing
- Usability Testing

What is GUI?

GUI stands for Graphical User Interface

It checks if all the basic elements are available on the page or not.

It involves checking the screens with the controls like menus, buttons,icons, and all types of bars like toolbar, menu bar, dialog boxes and windows etc.

It checks the spelling,alignments of the objects.

It checks if the mandatory fields are highlighted or not.It checks consistency in background color and color font type and font size etc.

What is Functional Testing?

Functional testing mainly involves black box testing.

It is not concerned about the source code of the application.

The prime objective of Functional testing is checking the functional behaviour of the software system.

It mainly concentrates on -

Mainline functions: Testing the main functions of an application
Basic Usability: It involves basic usability testing of the system. It checks whether a user can freely navigate through the screens without any difficulties.
Accessibility: Checks the accessibility of the system for the user

- Input Domain Coverage
- Database Testing
- Calculations/Manipulations
- Error Handling
- Links Existence & Links Execution
- Cookies & Sessions

Input Domain Coverage:

During input domain coverage test engineers validate data provided to the application with respect to value and length
There are two techniques in Input domain Techniques.

Equivalence Class Partition (ECP)
Boundary Value Analysis(BVA)

Database Testing:

During Database testing test engineers validate data provided to the application with respect to the database.

Validates DML operations(Insert , Update, Delete & Select)

Calculations/Manipulations:

Validating mathematical calculations

Error Handling:

Validate error messages thrown by the application when we provide invalid data.

The error messages should be clear and easy to understand to the user

Links Existence & Links Execution:

Links Existence - Links placed in the appropriate location or not.

Links Execution - Links are navigating to appropriate pages or not.

Cookies & Sessions:

Cookie- Temporary internet files which are created at client side when we open the websites. These files contain user data.

Session- Sessions are time slots which are allocated to the user at the server side.

Non-Functional testing:

Non-functional testing is a type of software testing to test non-functional parameters such as reliability,performance and compactability of the software.

The primary purpose of non-functional testing is to test the reading speed of the software system as per non-functional parameters.

Types of Non Functional Testing:

Performance testing

Security Testing

Recovery Testing

Compatibility Testing

Configuration Testing

Installation Testing

Sanitation Testing

Performance Testing:

Volume Testing-It refers to testing a software application with a certain amount of data. This amount can, in generic terms, be the database size or it could also be the size of an interface file.

Load Testing- Testing the speed of the system while increasing the volume gradually till where the system breaks.

Stress Testing-Testing the speed of the system suddenly increases and decreases the load to check where it's breaking.

Security Testing:

Testing security provided by the system.

Types:
Authentication
Access Control/Authorization
Encryption/Decryption

Compatibility Testing:

Testing the system performance in different OS and different browsers.

Usability Testing:

Usability Testing, also known as User Experience(UX) Testing, is a testing method for measuring an application how easy and user-friendly.

It mainly focuses on user using application, flexibility of application to handle controls and ability of application to meet its objectives.

User Acceptance Testing:

After completion of system testing the UAT. Testers collect all live data from clients. Last stage and higher level of testing.

Acceptance testing in two levels.

- Alpha testing - in company environment
- Beta testing - in client environment

Testing Terminology:

Smoke Testing

Sanity Testing

Regression Testing

Retesting

Adhoc Testing

Testing Terminology (conti)

Monkey Testing

End-End Testing

Exploratory Testing

Globalization Testing

Smoke Testing- It is the initial level of testing. It is used to check the stability of the build. It is also called build verification testing.
Sanity Testing- Testing the major functionalities of the build in depth. It is a subset of regression testing.
Regression testing- This testing to make sure that new code changes should not have side effects on the existing functionalities.
Re- Testing-Testing same test cases repetitively is called re-testing.
Adhoc Testing- Software testing performed without proper planning and documentation.
Testing is carried out with the knowledge of the tester about the application and the tester tests randomly without following the requirements.

Monkey Testing- Test the functionality randomly without knowledge of application and test cases is called Monkey Testing.

End to End Testing- Testing the overall functionalities of the system including the data integration among all the modules is called end-to-end testing.

Exploratory Testing- Exploring the application and understanding the functionalities adding or modifying the existing test cases for better testing is called exploratory testing.

Globalization Testing- Checks if the application has a provision of setting and changing languages date and time format and currency etc. If it is designed for global users, it is called globalization testing.It is also called as Internationalization Testing

SDLC Models or Testing Methodologies

Waterfall Model

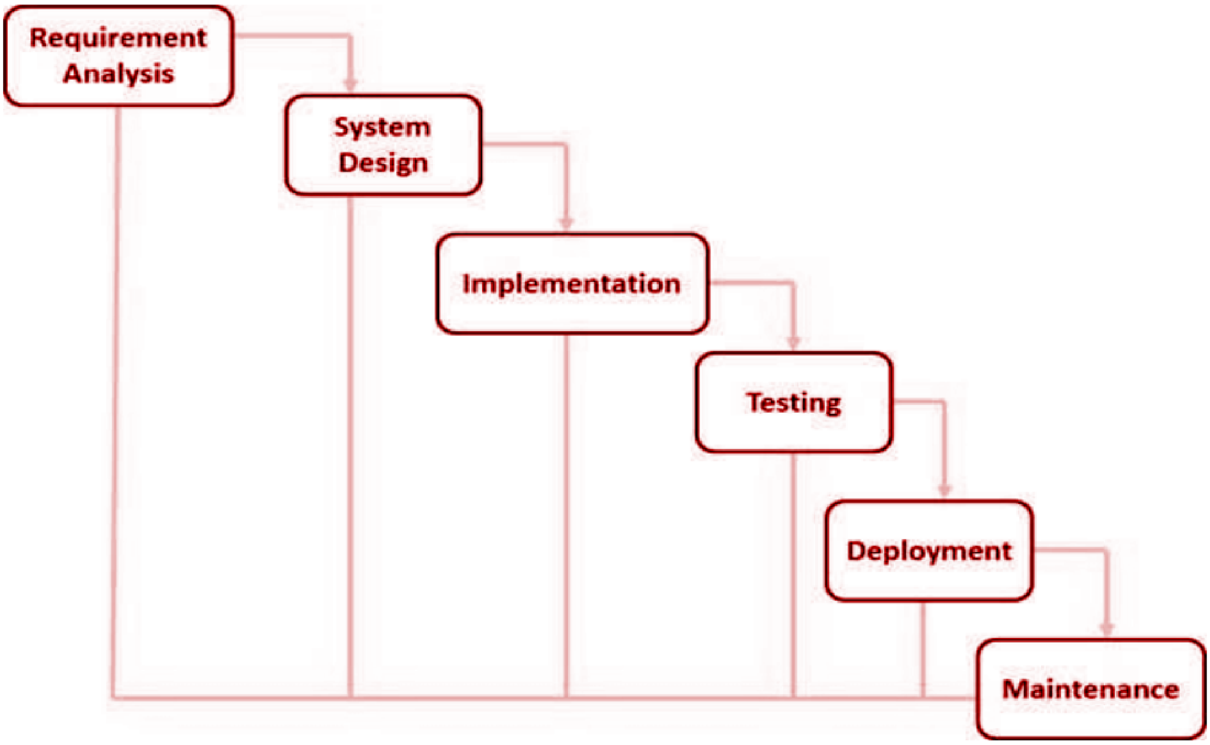
Incremental Model

Spiral Model

V-Model

Agile Model

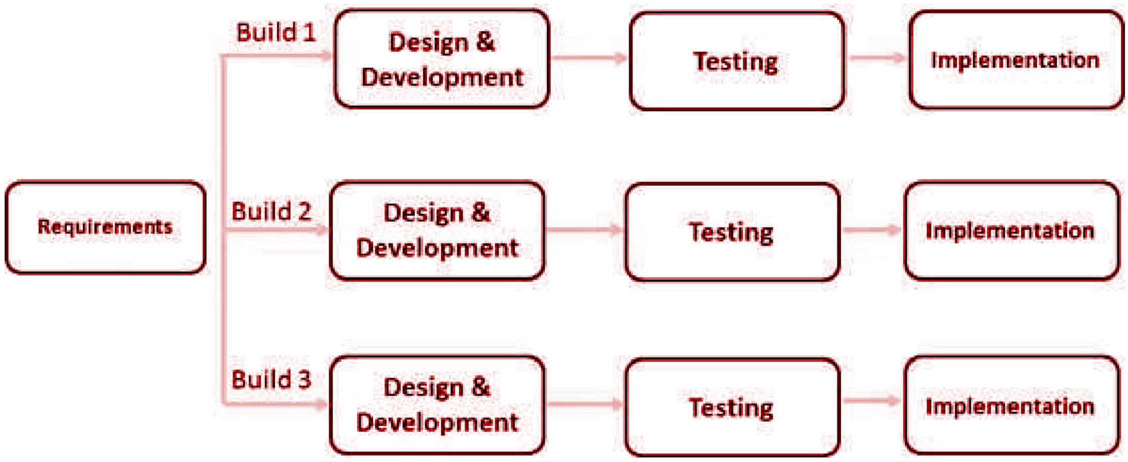
Waterfall model in SDLC



Waterfall model is a sequential model that divides software development into pre-defined phases. Each phase must be completed before the next phase can begin with no overlap between the phases.Requirements are not changing frequently

- Application is not complicated and big
- Project is short
- Requirement is clear
- Environment is stable
- Technology and tools used are not dynamic and is stable
- Resources are available and trained

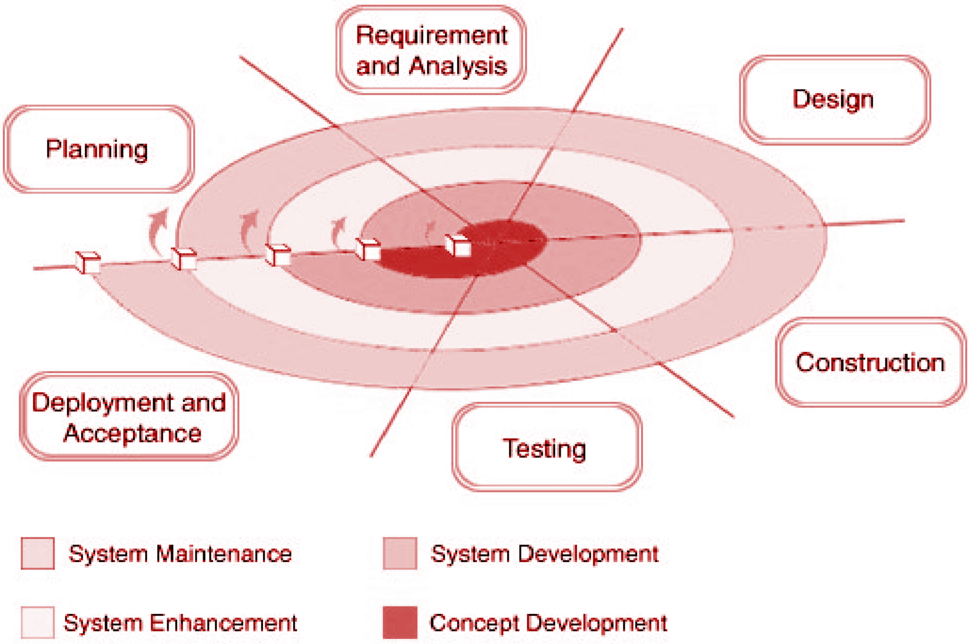
Incremental Model in SDLC



Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of the software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance. The incremental model is not a separate model. It is essentially a series of waterfall cycles.

- When demand for an early release of a product arises
- When high-risk features and goals are involved
- Such methodology is more in use for web application and product based companies

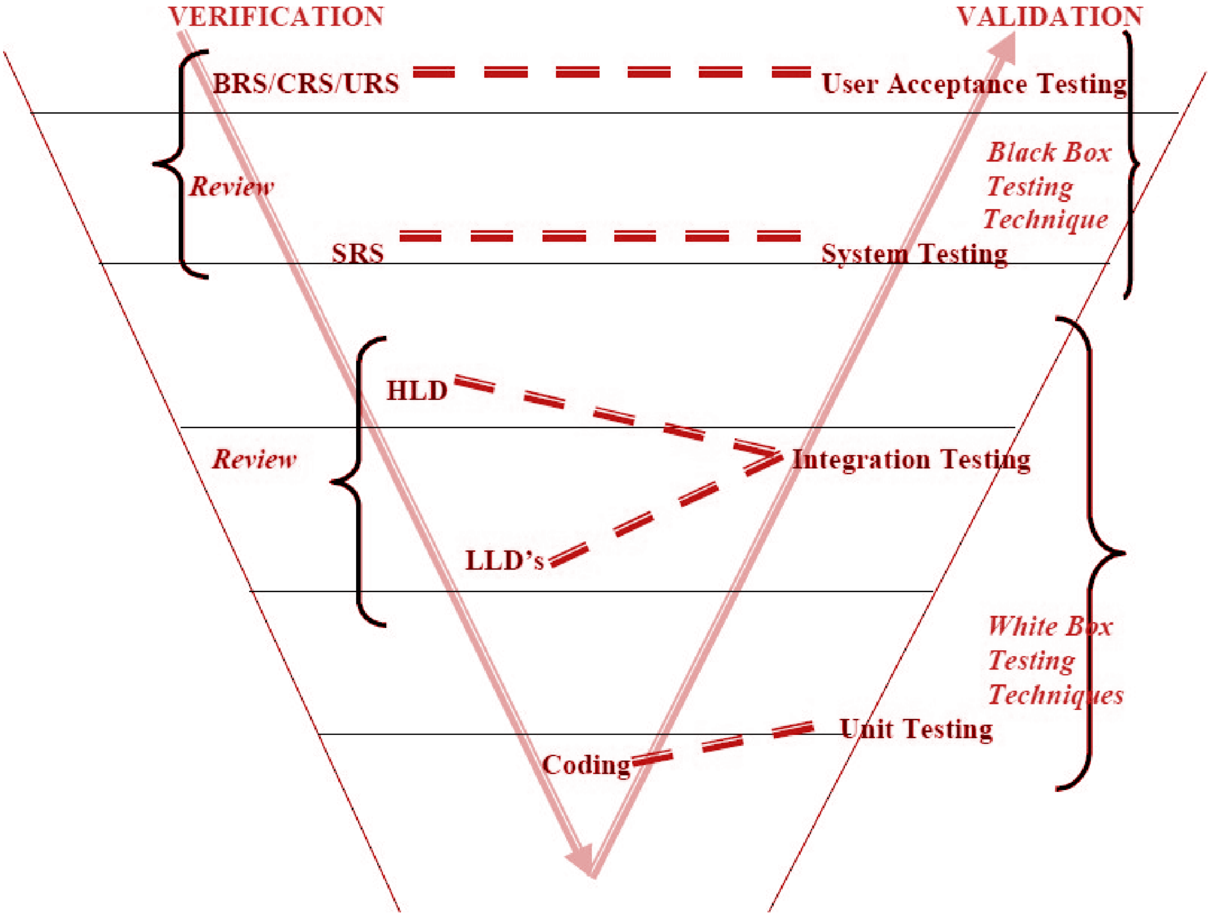
Spiral Model



Spiral Model is a risk-driven SDLC model. It is a combination of waterfall model and iterative model. Spiral Model helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process.

- When requirements are unclear and complex, Spiral model in SDLC is useful
- When releases are required to be frequent, spiral methodology is used
- When changes may require at any time
- Spiral methodology is useful for medium to high-risk projects

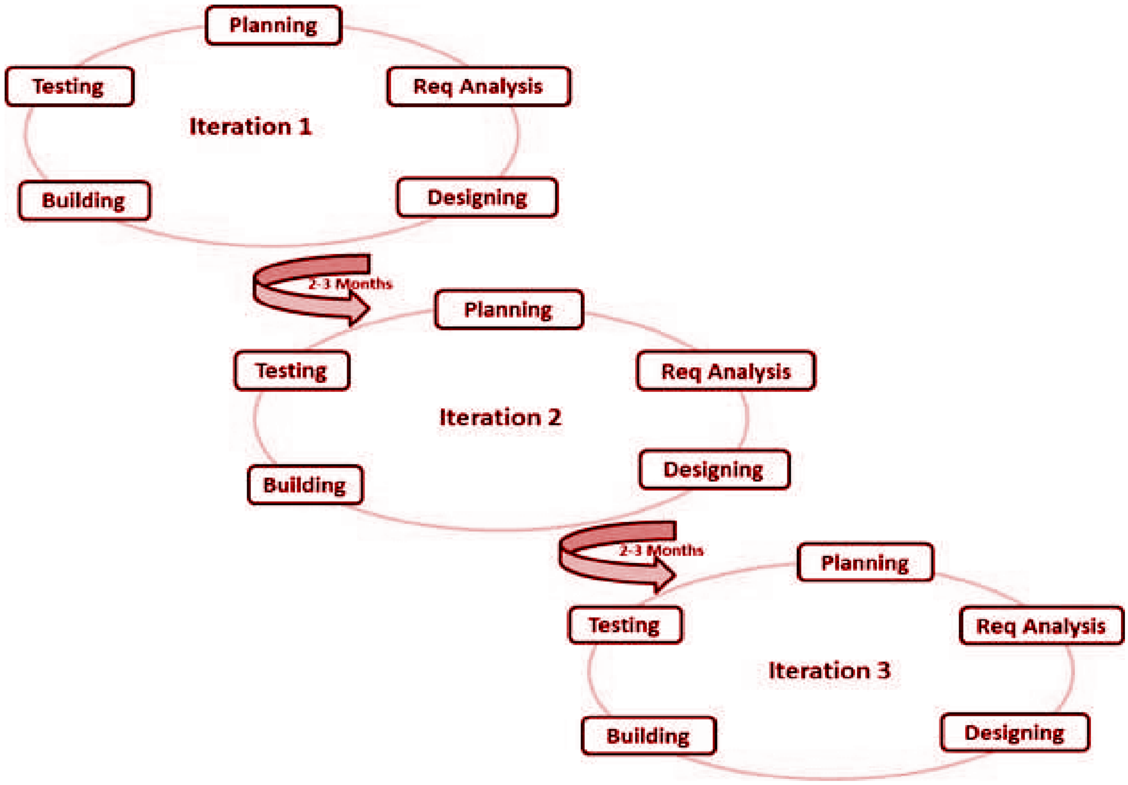
V-Model in SDLC



V Model is a highly disciplined SDLC model in which there is a testing phase parallel to each development phase. The V model is an extension of the waterfall model in which testing is done on each stage parallel with development in a sequential way. It is known as the Validation or Verification Model.

- Testing activities like planning, test designing happens well before coding
- Proactive defect tracking – that is defects are found at early stage
- Easy and simple to use.

Agile Model in SDLC



Agile methodology is a practice which promotes continue iteration of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.

- Face-to-Face Communication with clients.
- When a customer is ready to have a meeting with a software team all the time.
- Frequent Delivery.
- It reduces total development time.
- Anytime changes are acceptable.

Other Testing Concepts

Defect Leakage- After product release Client or end user identifies any bug then it is called as **defect leakage** or **bug leakage**.

Defect Release- Defect or Bug release is when software or an application is handed over to the client knowing that the defect is present in a particular release.

Defect Triage- Defect triage is a process where each bug is prioritized based on its severity, frequency, risk, etc. It is done by higher managerial level Project Manager,Test Team Leader,Technical Lead,Development Team Leader.

Testware- Testware is test artifacts like test cases, test data, test plans needed to design and execute a test.

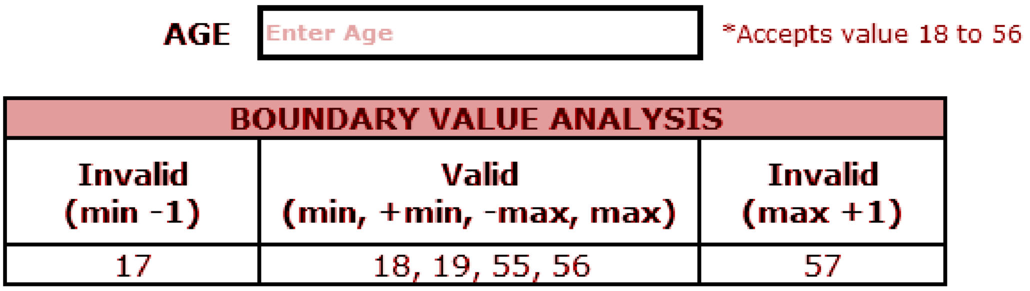
Test case Design Techniques

It is also called as blackbox design techniques
It is also called as validation techniques

- Boundary Value Analysis (BVA)
- Equivalence Class Partitioning (ECP)
- Decision Table Testing
- State Transition Table

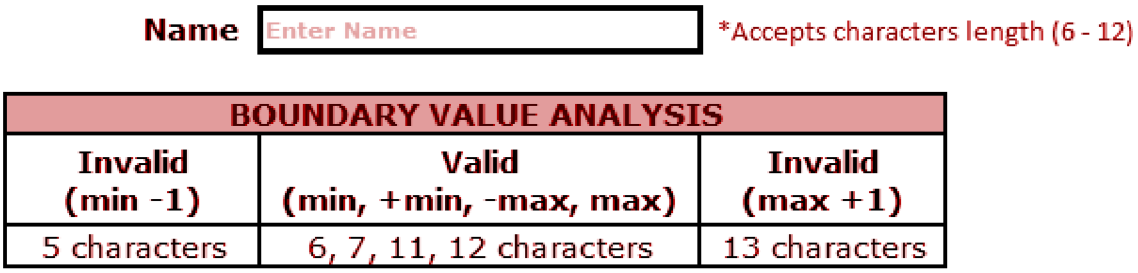
Boundary Value Analysis (BVA)

Boundary value analysis (BVA) is based on testing the boundary values of valid and invalid partitions. The Behavior at the edge of each equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects.



AGE Enter Age *Accepts value 18 to 56

BOUNDARY VALUE ANALYSIS		
Invalid (min -1)	Valid (min, +min, -max, max)	Invalid (max +1)
17	18, 19, 55, 56	57

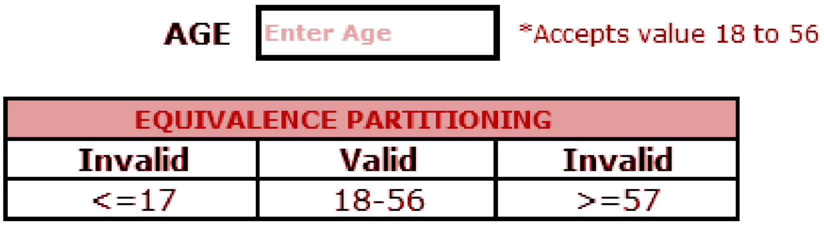


Name Enter Name *Accepts characters length (6 - 12)

BOUNDARY VALUE ANALYSIS		
Invalid (min -1)	Valid (min, +min, -max, max)	Invalid (max +1)
5 characters	6, 7, 11, 12 characters	13 characters

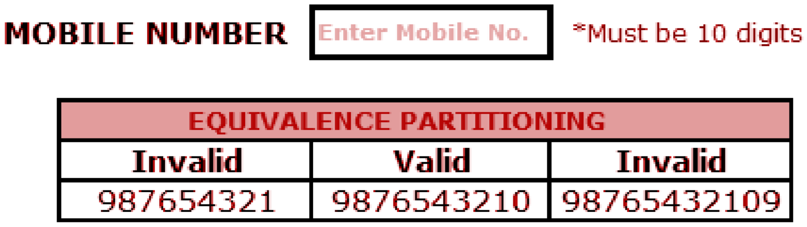
Equivalence Class Partitioning (ECP)

In equivalence partitioning, inputs to the software or system are divided into groups that are expected to exhibit similar behavior. Hence selecting one input from each group to design the test cases.



AGE Enter Age *Accepts value 18 to 56

EQUIVALENCE PARTITIONING		
Invalid	Valid	Invalid
<=17	18-56	>=57



MOBILE NUMBER Enter Mobile No. *Must be 10 digits

EQUIVALENCE PARTITIONING		
Invalid	Valid	Invalid
987654321	9876543210	98765432109

Testers Roles and Responsibilities

Roles:

Validating the Application

Identifying the bugs

Ensure the good quality product to client

Responsibilities:

Analysing requirement

Identifying Test scenario

Writing test case

Executing test cases

Identifying defects

Deals with defect tracking process

Updating RTM

Attending day to day meetings(Scrum call, Sprint Planning,Sprint review,)