

# Meeting Etiquette

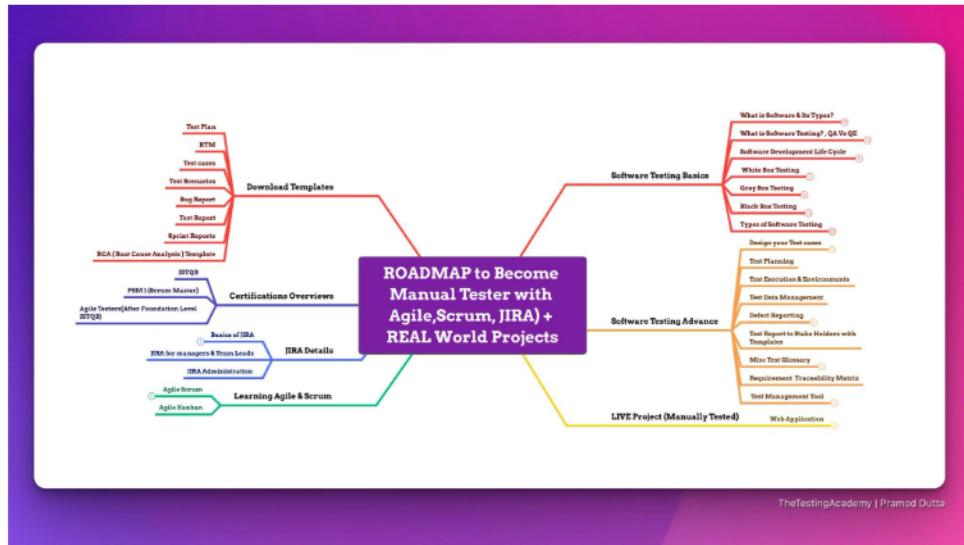
- Please be on Mute all the time.
- Turn off Video, So that we can save bandwidth.
- There is separate Q&A Section in End Please use that, Add questions in Google Form.
- Break at 5 min.
- Mute your microphone
- To help keep background noise to a minimum, make sure you mute your microphone when you are not speaking.
- Be mindful of background noise
- Avoid multitasking
- All links and Slides will be shared.



# { Software Tester } BluePrint.



Pramod Dutta



TheTestingAcademy | Pramod Dutta

Exact Blueprint.  
You Need to Become Software Tester.

# Agenda

- Introduction to Software Testing?
- Details on SDLC and Different Models.
- STLC Life Cycle & 7 Principles of Software Testing.
- RTM & Different Types of Software Testing.
- Test Design Techniques
- Bugs, Severity vs Priority



# Rules

**Focus on One  
Thing.**

**5% : 95% Rule**

**70% is Perfect  
100% is Failure**

**New Action**



**Commitment!**  
**Block at least**  
**3-4 hour Per Week.**

# What is Software?

Software is basically a set of instructions or commands that tells a computer what to do

Windows Calculator



# What is Software?



ment Web App



Word Online



Gmail



Gmail Offline



Google Docs



Mobile Websi



look.com



Google Drive



Box



FollowMania



YouTube



Daum Equatio



ihoo Wiki



Photo Book



PDF to Word Converter...



SnapPages



Sticky Notes



SAPOMe

# Types of Softwares

- **System software**

Ex: Device drivers, Operating Systems, Servers, Utilities, etc.

- **Programming software**

Ex: compilers, debuggers, interpreters, etc.

- **Application software**

Ex: Web Applications, Mobile Apps, Desktop Applications etc

# Types of Software

## System Software

System software basically controls a computer's internal functioning and also controls hardware devices such as monitors, printers, and storage devices, etc

Operating System  
Language Processor  
Device Driver  
Written in a low-level language

**Driver software.** ...  
**Middleware.** ...  
**Programming software.**

## Application Software

Application software is designed to perform a specific task for end-users

It is a product or a program that is designed only to fulfill end-users' requirements

General Purpose Software  
MS-Word, MS-Excel, PowerPoint, etc.  
Customized Software  
airline reservation system  
Utility Software  
antivirus, disk fragmenter, memory tester, disk repair, disk cleaners  
Written in a high-level language

# What is Software Testing

**Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do.**

**Software Testing is a part of software development process.**

Main Objective of testing is to release quality product to the client.

**Activity to detect and identify the defects in the Software. (Not Fix)**

# What are software testing objectives and purpose?

# What are software testing objectives and purpose?

To prevent defects.

## Finding defects

Gaining confidence in and providing information about the level of quality

ensure that it satisfies the BRS that is Business Requirement

end result meets the business and user requirements.

gain the confidence of the customers by providing them a quality product.



# Quality

**Quality is defined as justification of all the requirements of a customer in a product.**

quality  
'क्यूलिटी

noun

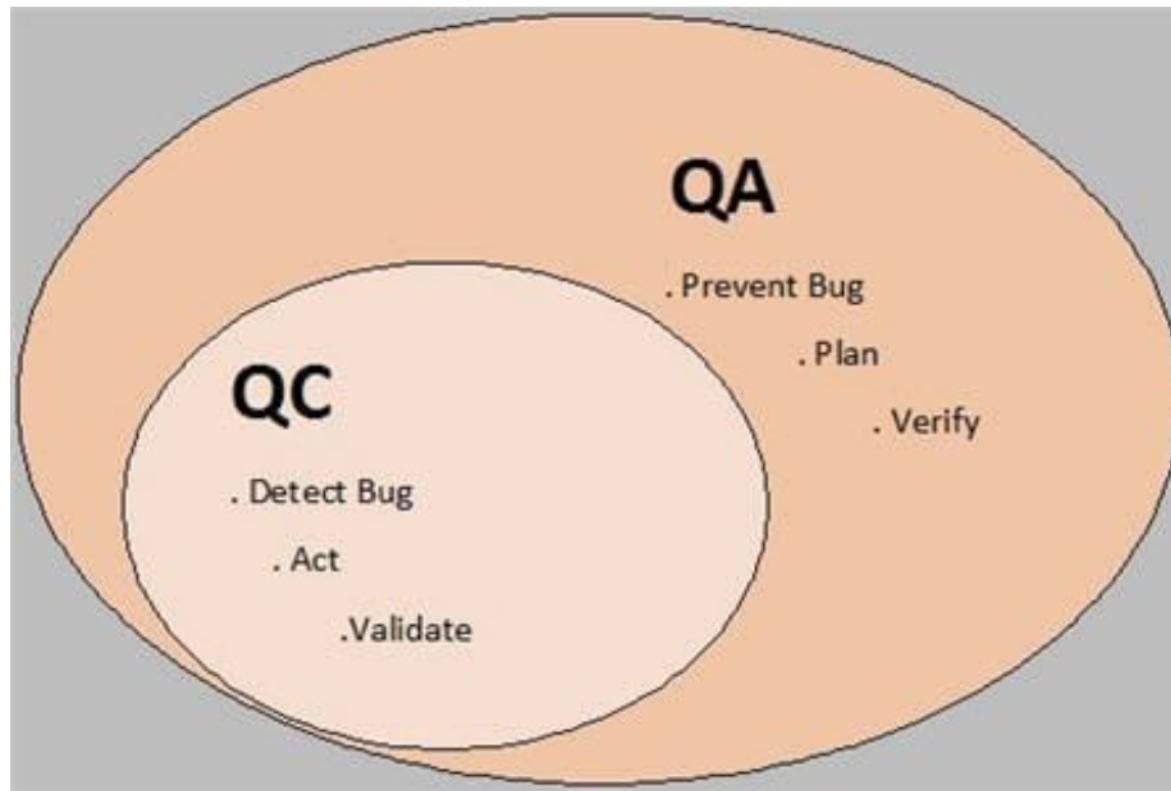
1. how good or bad something is  
(किसी वस्तु की) उत्कृष्टता का स्तर (कम या अधिक); गुणवत्ता  
to be of good/poor/top quality ↗
2. a high standard or level  
उच्च स्तर, दर्जा या कोटि  
Aim for quality rather than quantity in your writing. ↗

TheTestingAcademy | Pramod Dutta

## Quality software is Means

- Bug-free
- Delivered on time.
- Within budget.
- Meets requirements and/or expectations.
- Maintainable

Quality Assurance (QA)	Quality Engineer (QE)	Software Development Engineer in Test (SDET)
QA ought to be familiar with the bug tracking, ticketing, and testing processes.	QE must be familiar with operations as infrastructure, servers, platforms, etc.	SDET must be able to do advanced automated tasks.
Software engineering technical expertise, such as SQL overload or basic programming, is required by QA.	Security testing, performance testing, and integrating checks in a CI/CD methodology are all skills that QE should have.	SDET ought to be able to perform white box testing.
Manual and automated testing in Selenium, Cucumber, SoapUI, JMeter, and other tools should be familiar to QA.	QE ought to be able to test automation at several levels, including API, UI, and protocol.	SDET must be well-versed in development.
QA must be able to ask the proper questions, listen carefully to replies, thoroughly explain issues, and perform well under pressure.	Selenium, Cucumber, SoapUI, JMeter, and other tools should be familiar to QE.	SDET should be able to create orchestration platforms.
	Quality should be a concern for QE.	



## Quality Assurance vs Testing

**Quality  
Assurance**

**Quality  
Control**

**Testing**

# Comparison between QA, QC and Testing

## Quality Assurance

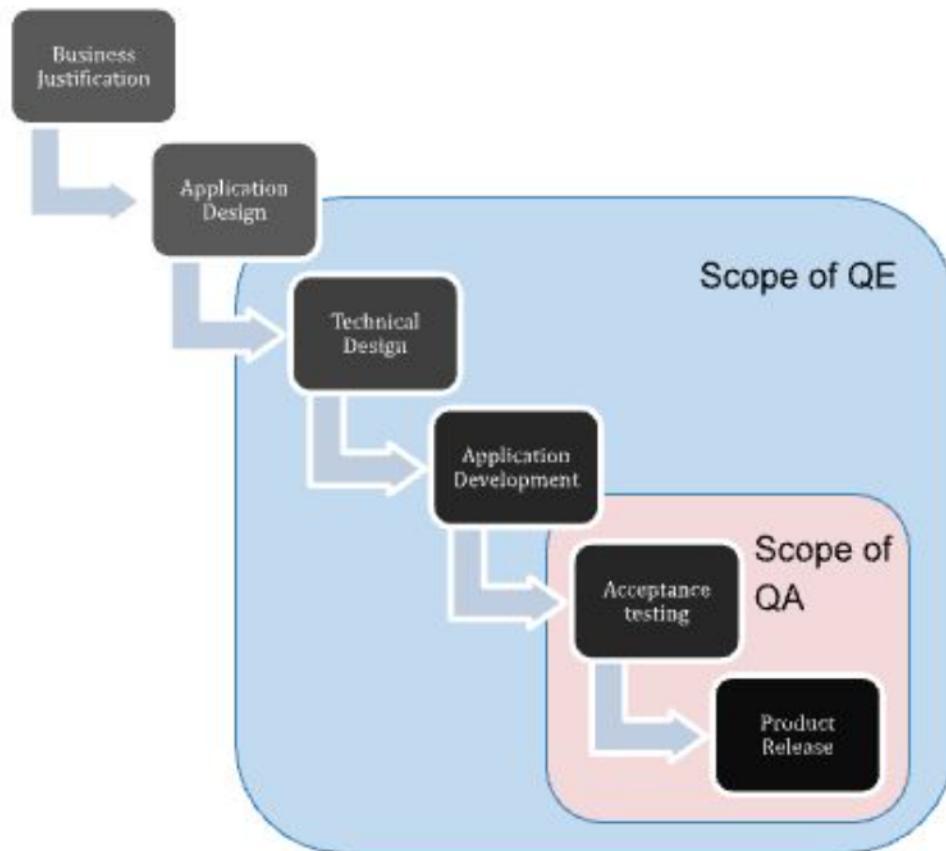
- Subset of SDLC
- Process oriented
- Ensure that processes and procedures are in place to achieve quality
- Focus on process to achieve required quality
- Prevent defects
- Whole team approach
- Proactive process

## Quality Control

- Subset of QA
- Product oriented
- Activities to ensure the product quality
- Focus on product to check for the required quality
- Find and fix defects
- Reactive process
- Testing team

## Testing

- Subset of QC
- Product oriented
- Validate the product against specifications
- Focus on actual testing of the product
- Find and fix defects
- Reactive process
- Testing team



# Software Testing

To check whether the **Actual** software product matches **Expected** requirements and to ensure that software product is Defect free.

Id	Input	Expected Result	Actual Result	Status
1	Read ATM Card	Accept card and ask for pin #	Accepted the card and asked for a pin.	Passed
2	Read Invalid Card	"No ATM Card" exception is thrown and card is returned to the user.	Accepted the card and asked for a pin.	Failed
3	Invalid PIN Entered	"Stolen Card" exception is thrown and card is destroyed.	Accepted the card and asked for a pin.	Failed

# Why Software Testing is Important?

## 4. Bitcoin Hack, Mt. Gox, 2011

Mt. Gox was the biggest bitcoin exchange in the world in the 2010s, until they were hit by a software error that ultimately proved fatal.

The [glitch](#) led to the exchange creating transactions that could never be fully redeemed, costing up to \$1.5 million in lost bitcoins.

But Mt. Gox's woes didn't end there. In 2014, they lost more than 850,000 bitcoins (valued at roughly half a billion USD at the time) in a hacking incident. Around 200,000 bitcoins were recovered, but the financial loss was still overwhelming and the exchange ended up [declaring bankruptcy](#).

<https://raygun.com/blog/costly-software-errors-history/>

# Why Software Testing is Important?

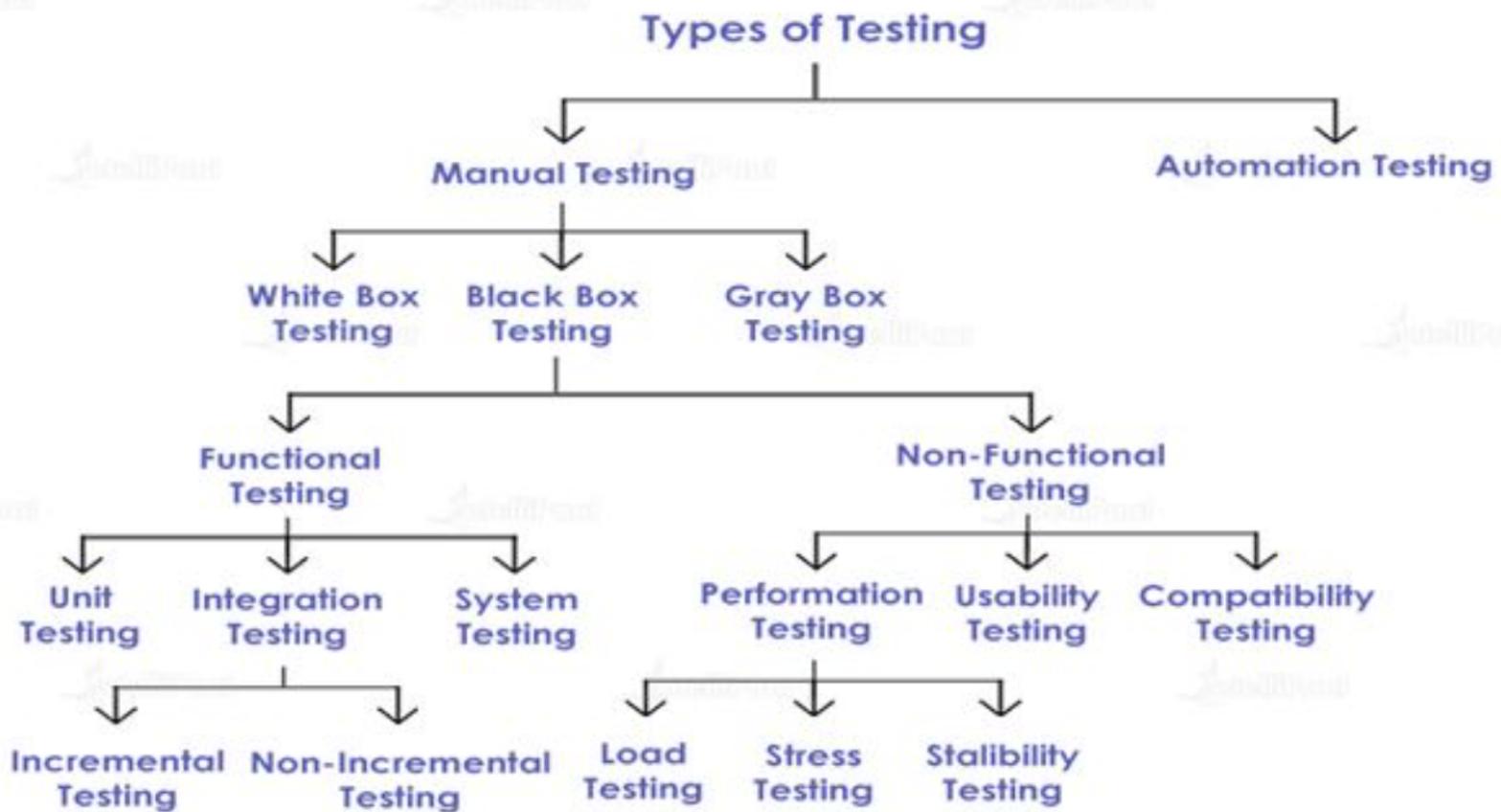
Vulnerability in Windows 10. This bug enables users to escape from security sandboxes through a flaw in the win32k system

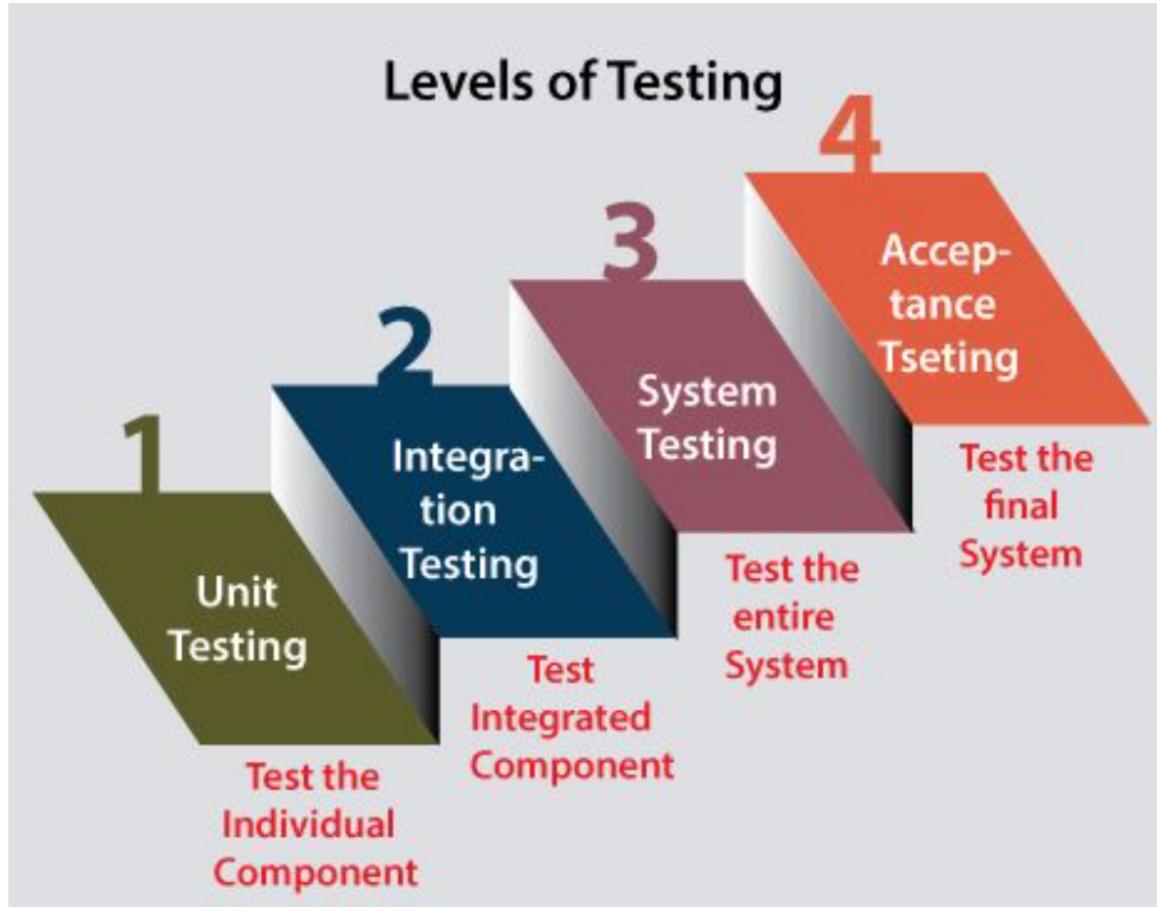
China Airlines Airbus A300 crashed due to a software bug on April 26, 1994, killing 264 innocents live

<https://raygun.com/blog/costly-software-errors-history/>

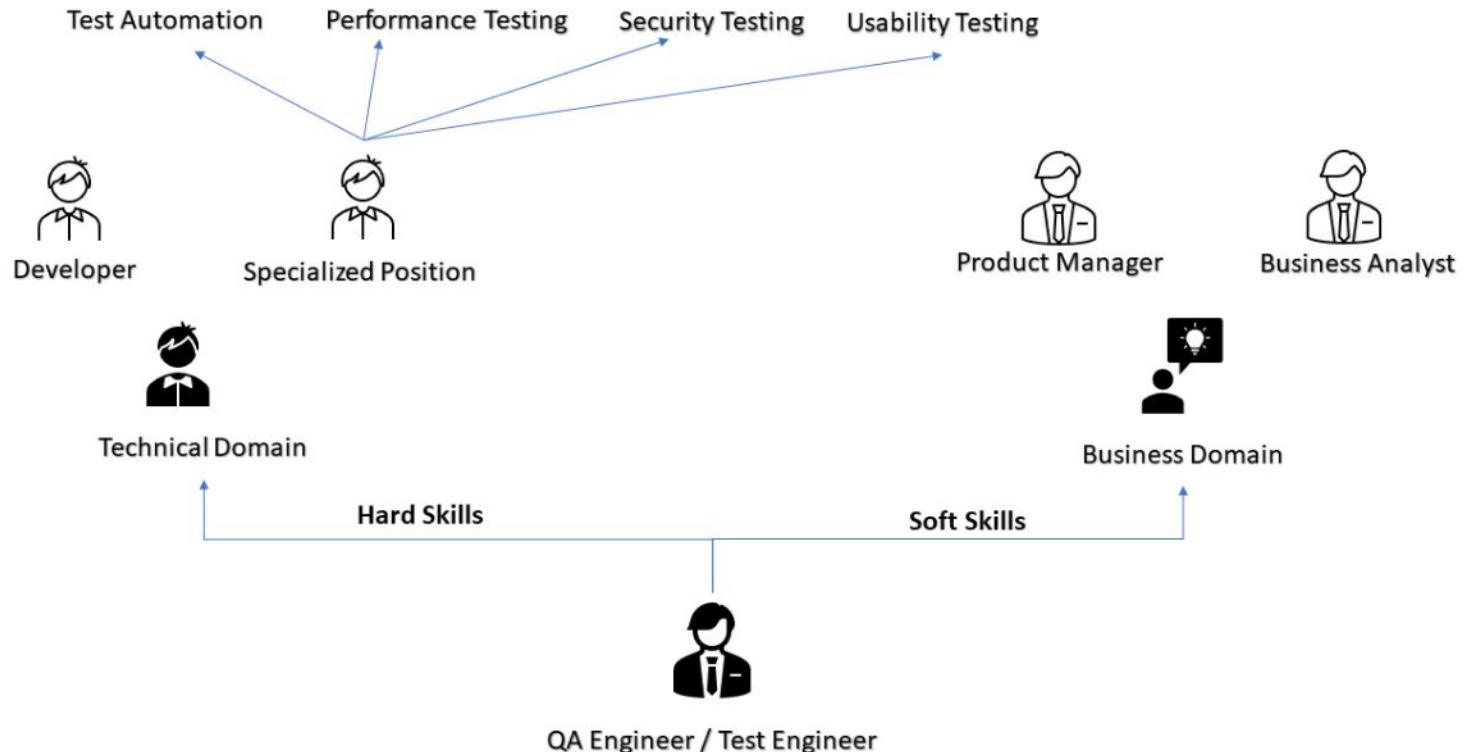
# Benefits of Software Testing

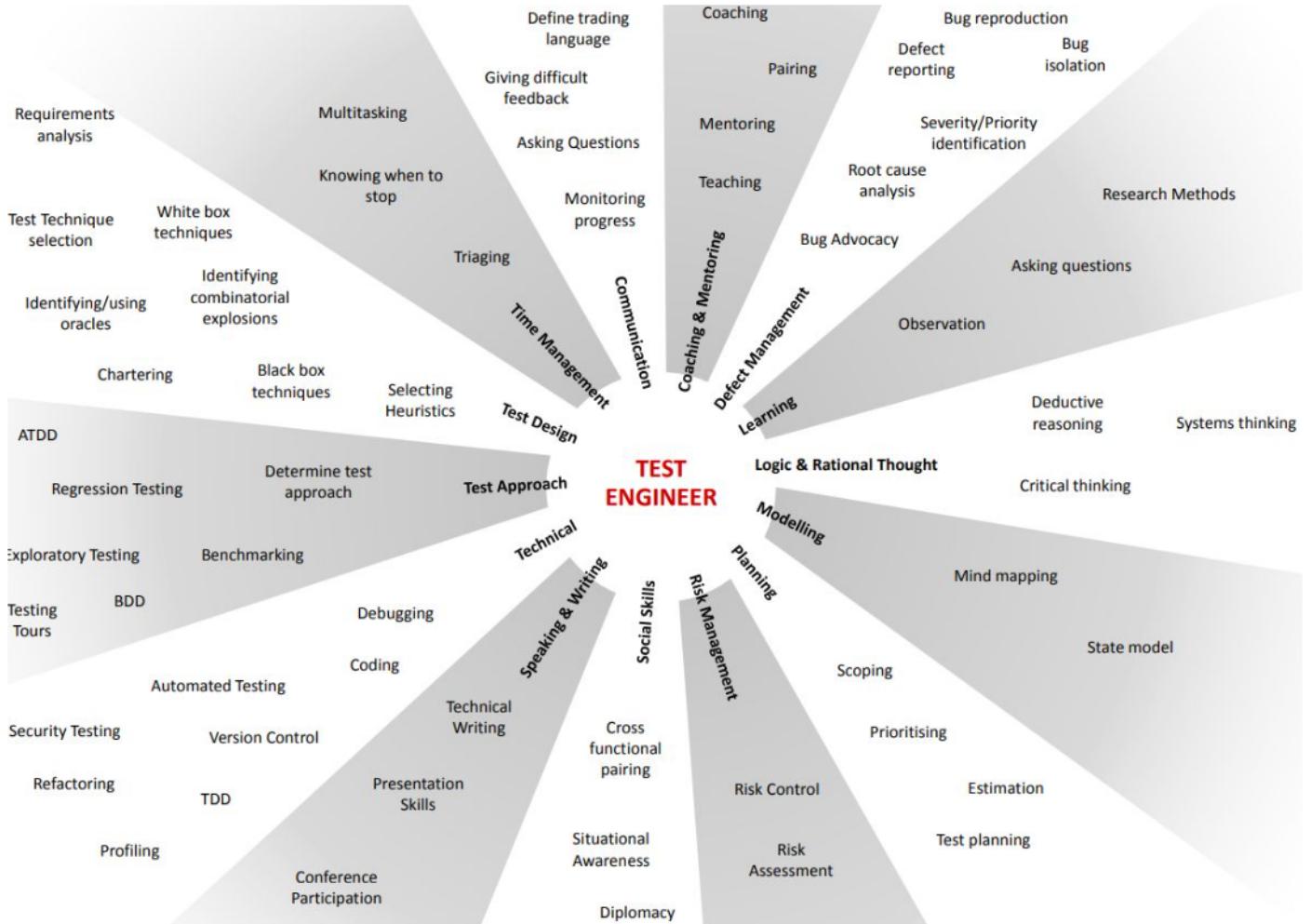
- Cost-Effective
- Product quality
- **Customer Satisfaction**









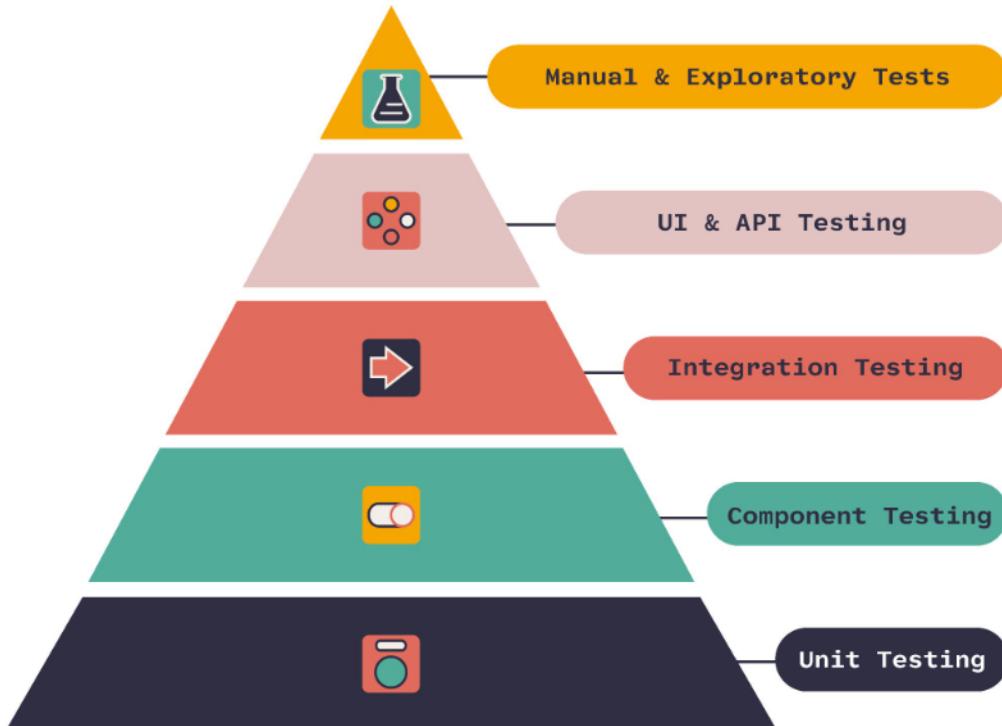


# Types of Software Testing

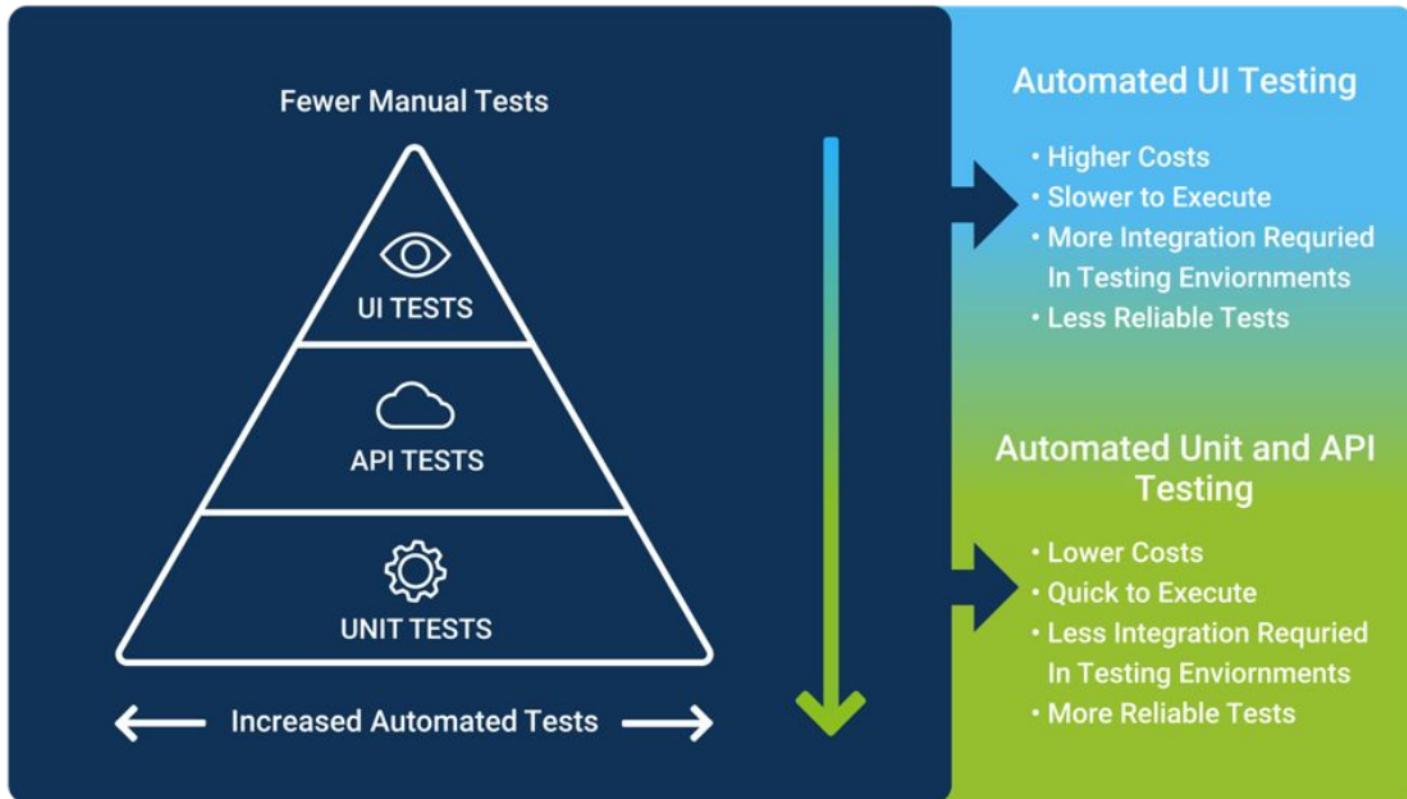
**Functional Testing**

**Non-Functional Testing**

# Testing Pyramid



# The Automation Pyramid



# Unit Testing

Type of software testing where **individual units** or components of a software are tested

Unit tests help to fix bugs early in the development cycle and save costs.

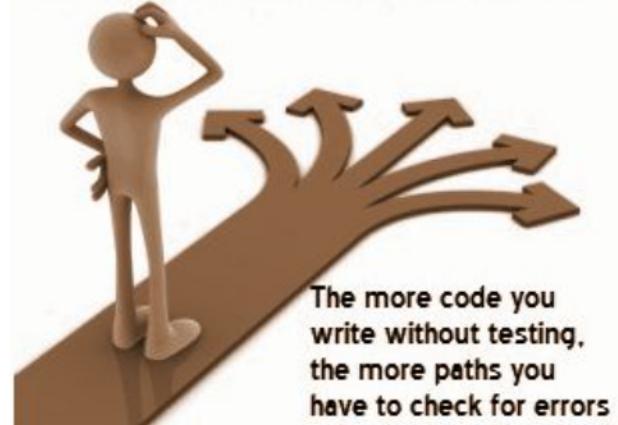
A developer writes a section of code in the application just to test the function. They would later comment out and finally remove the test code when the application is deployed.

A coder generally uses a UnitTest Framework to develop automated test cases

# Code coverage techniques used in Unit Testing

- Statement Coverage
- Decision Coverage
- Branch Coverage
- Condition Coverage
- Finite State Machine Coverage

Keep on a straight path with proper unit testing.

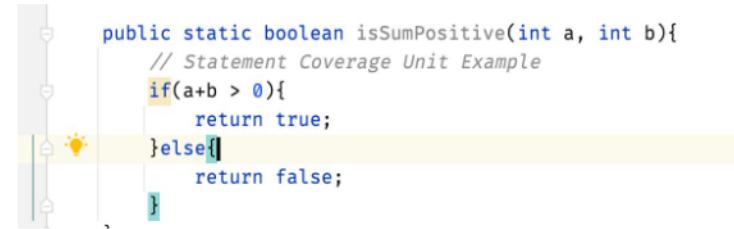


# Unit Testing

## Demo of Web App unit

isSumPositive function

```
@Test  
public void shouldAnswerWithPass()  
{  
    Assert.assertTrue(AddModule.isSumPositive( a: 2, b: 3));  
}  
  
@Test  
public void shouldAnswerWithFail()  
{  
    Assert.assertTrue(AddModule.isSumPositive( a: 2, b: -3));  
    //Assert.assertFalse(AddModule.isSumPositive(2,-3));  
}
```



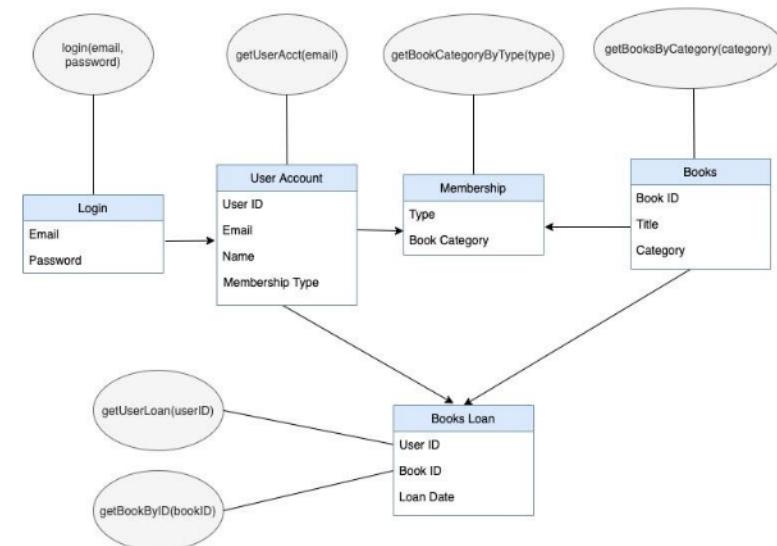
```
public static boolean isSumPositive(int a, int b){  
    // Statement Coverage Unit Example  
    if(a+b > 0){  
        return true;  
    }else{  
        return false;  
    }  
}
```

# Integration Testing

Type of software testing where individual units or components of a software are tested

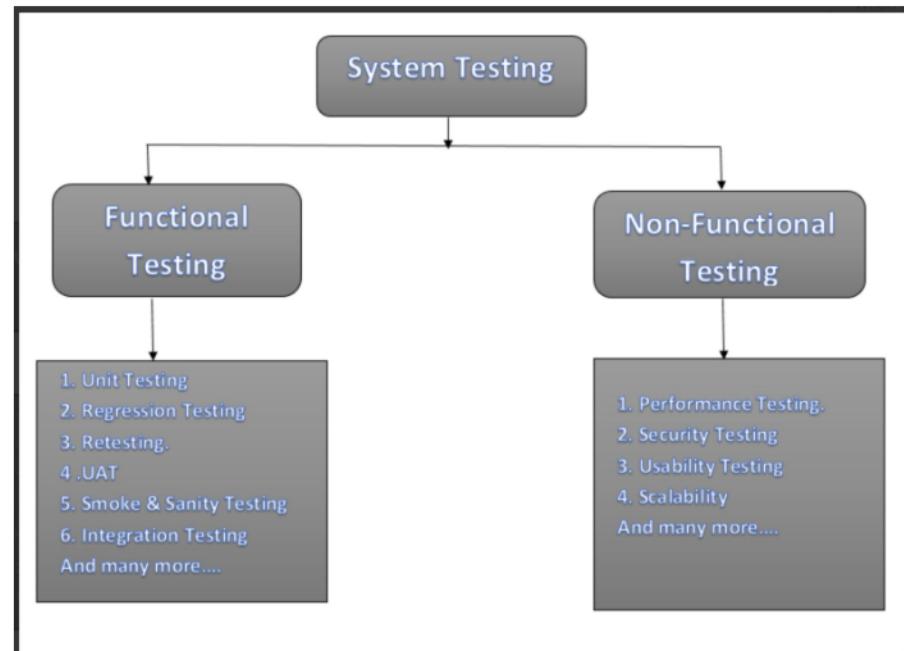
Software modules are integrated logically and tested as a group.

Testing flow of data/information between the modules.



# System Testing

System testing is a testing level in which tests are performed to know if a complete build aligns with functional and nonfunctional requirements made for it



# System Testing vs Integration Testing



	System Tests	Integration Tests
Intention	To guarantee that the total build fulfills the business specifications.	To guarantee that joined units can act together without problems.
Type	Nonfunctional and functional type of test. It falls in the acceptance testing class.	Functional type of test. It's not in the acceptance testing class.
Technique	Black box testing	White and black box testing or gray box testing
Level	Three (3)	Two (2)
Value	Helps to identify system errors.	Helps to identify interface errors.
Teams involved	Developers and Testers	QA

# System Testing vs E2E Testing

## System Testing

It is carried out once integration testing is performed.

## End-to-end Testing

It is performed after the system testing.

# Levels of Testing



## UNIT TESTING

Test Individual Component

## INTEGRATION TESTING

Test Integrated Component

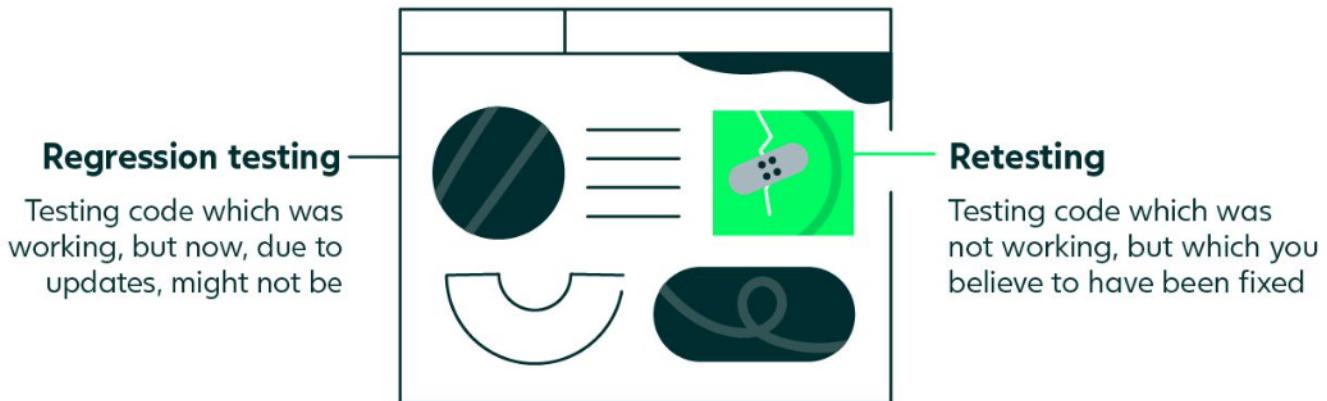
## SYSTEM TESTING

+ Test the entire System

## ACCEPTANCE TESTING

Test the final System

# Regression testing vs. retesting



## Regression Testing

VS

## Re-Testing

Focuses on both failed and successful test cases.



Focuses only on failed test cases.

Test cases can be automated.



Test cases can't be automated.

Verifies whether any change has broken the existing functionalities.



Reveals whether a fix causes a special defect in the application.

Priority of regression testing is lower than retesting so executed in parallel.



The priority of retesting is higher than regression testing so executed first.

It is carried out for defects in general.



It is carried out for specific defects.

Test cases can be obtained before starting the testing process.



Test cases can't be obtained before starting the testing process.

# Manual Tester Roles and Responsibilities

- Gather requirements from team
- Prepare Test Plan, Test Scenarios, Test cases -  
<https://sdet.live/3DbA>
- Verifying the Software Web/ App by Hand
- Execute the Test cases and Report Bugs
- Send Test report to stakeholders
- More concentrated on the UI/UX issues
- In charge of paperwork(aka documentation online)

# Manual Tester Roles and Responsibilities

- Test environment setup
- Participation in meetings
- Analysis of customer requests
- Software bug tracking
- Analysis and execution of test cases
- Maintaining contact with test managers

# What you do can extra? As Manual tester

- Help in preparing the requirements to PM
- Share a Video or Images of the Manual flows to automation team, so that they can create better automation.
- Help the team pm, devs to sync and come with timelines of release.
- Learn coding and help in automation of the flows manual tested.
- Identify the automation flows and pain manual areas automate them using scripts

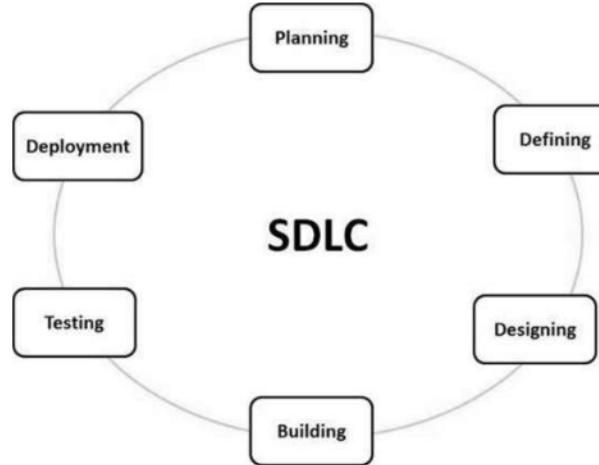
# What you do can extra? As Manual tester

<https://forms.gle/KcCe2bhZSFsS5GzLA>

# 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.

ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.



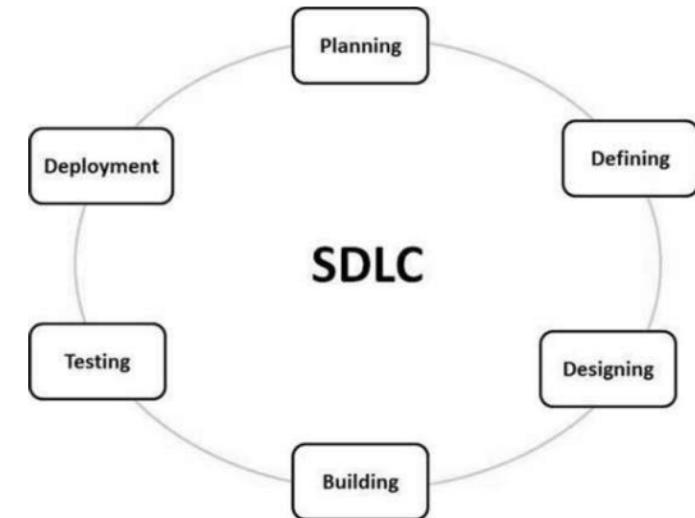
# 1. Planning and Requirement Analysis

- It is performed by the senior members of the team with inputs from the customer.
- Sales department, market surveys and domain experts in the industry.

Documents - Statement of Work, Project Plan

Outcome - Various Technical approaches that can be followed to implement the project successfully with minimum risks.

<https://bugz.atlassian.net/l/cp/EeXpfJOW>



# SOW-TEMPLATE-Project Manager-ND

[https://docs.google.com/document/d/1oP7Fw3RTvCdYuc1sH\\_1MeRYzOUhmtapr/edit?usp=sharing&ouid=104755920778477387077&rtpof=true&sd=true](https://docs.google.com/document/d/1oP7Fw3RTvCdYuc1sH_1MeRYzOUhmtapr/edit?usp=sharing&ouid=104755920778477387077&rtpof=true&sd=true)

# 2. Defining Requirements

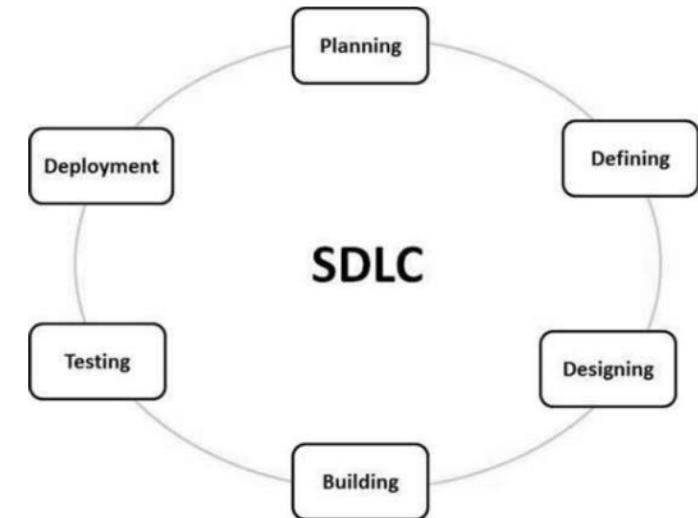
- Define and document the product requirements and get them approved from the customer or the market analysts

Documents - SRS (Software Requirement Specification) document

<https://sdet.live/samplesrs>

Consists of all the product requirements to be designed and developed during the project life cycle.

<https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>



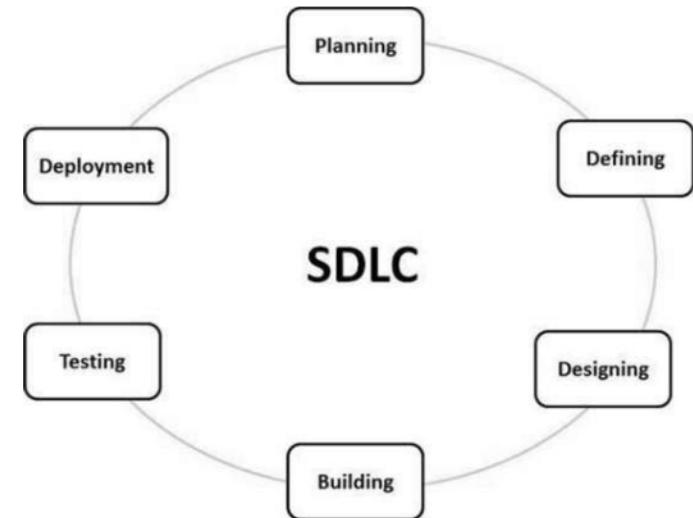
# 3. Designing

- Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification

Documents - DDS

<https://sdet.live/samplesrs>

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation



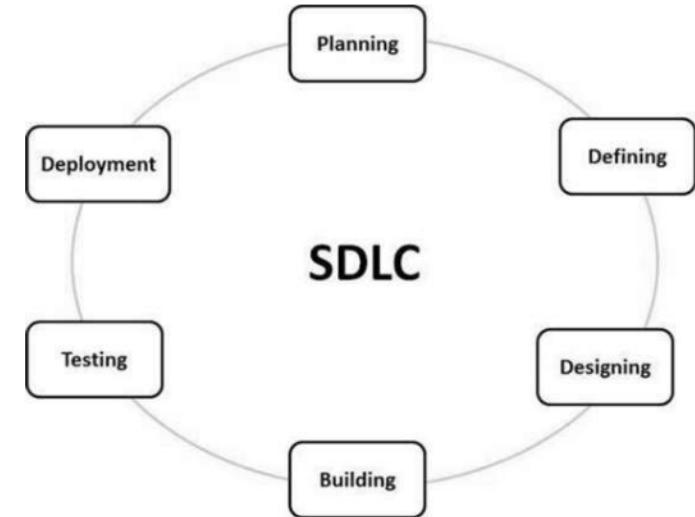
# 4. Building

- The programming code is generated as per DDS during this stage

Documents - FRDs

<https://sdet.live/samplesrs>

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers



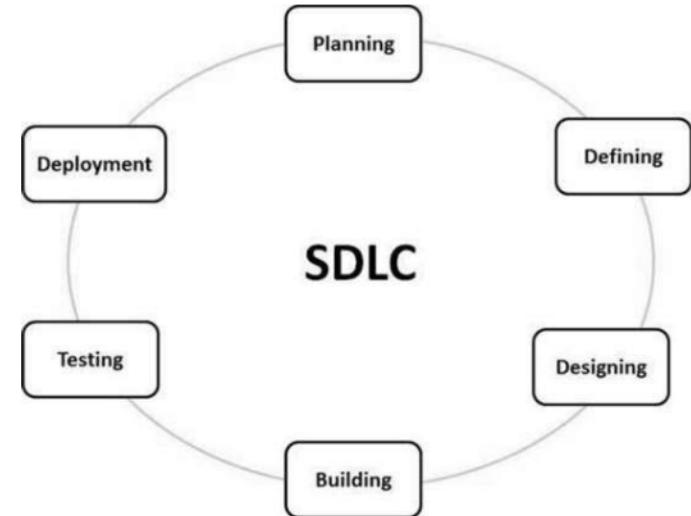
# 5. Testing

- This stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Documents - multiple Docs

<https://sdet.live.notes>

Full STLC Life Cycle

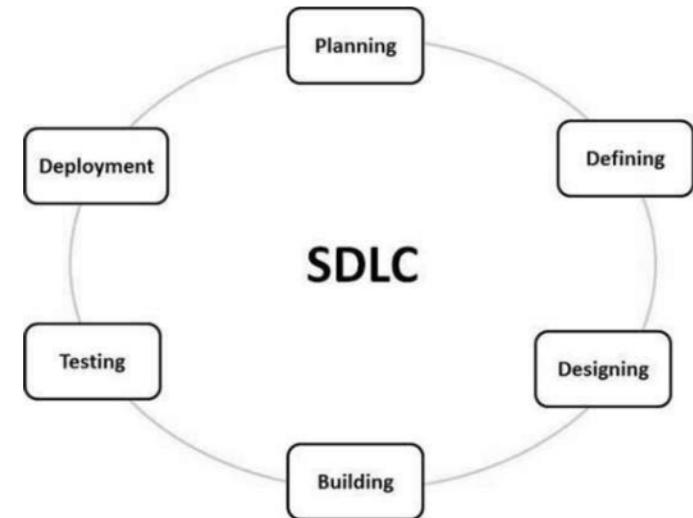


# 6. Deployment

- Once the product is tested and ready to be deployed it is released formally in the appropriate market.

Documents - NA

The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).



# BRD

**Business Requirement Document i.e. BRD** is created during the initial phase of the project. This document contains the high-level business requirements that could be easy to follow by business stakeholders, managers, board of directors, etc.

a business requirement document is a functional business requirement that is written in a well-structured manner without any technical jargon.

- Bank customers should be able to register themselves.
- Registered bank customers should be able to log in.
- Customers should be able to do online transactions.
- Customers should be able to open fixed deposit online.
- Customers should be able to recharge their mobile phones, etc.

# FRD

**Functional specification document** is required with detailed requirements in technical terms that will be referred by the technical team for further development of the system.

- Username should not include numeric value.
- The password should be 8 or more char long.
- Only 5 transactions should be allowed in a day, etc

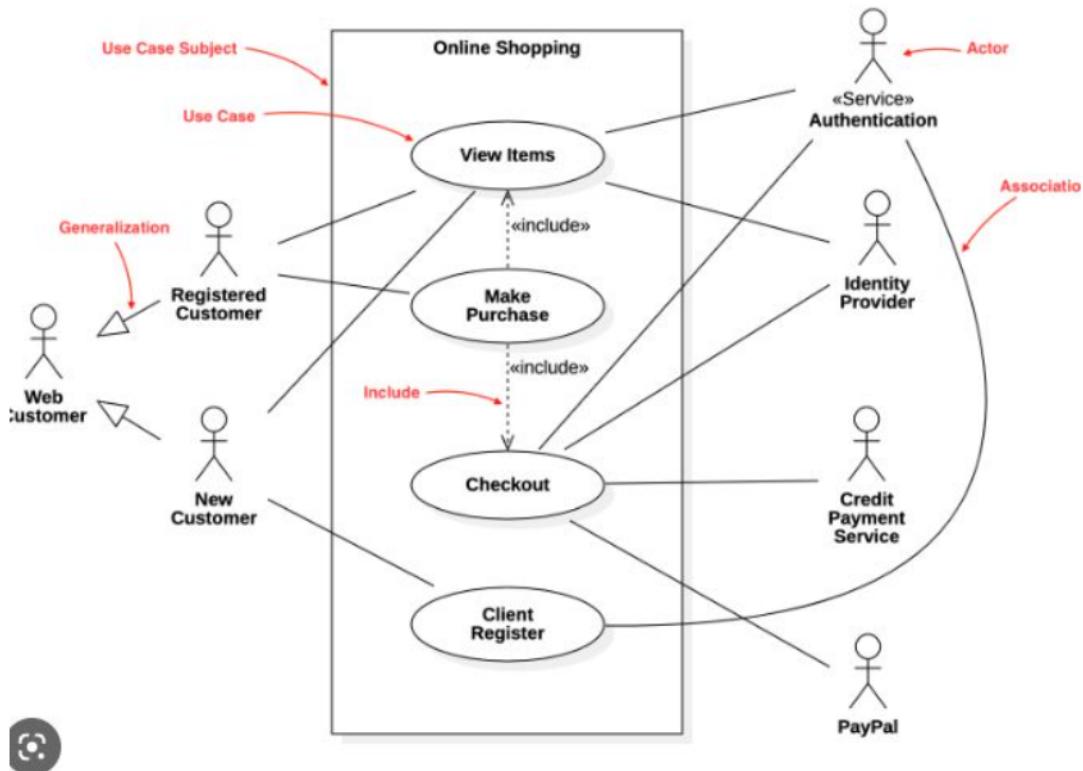
# SRS

**Software Requirement Specification** i.e. SRS Document is one of the important documents for the development team. It is a complete description of the behavior of a system to be developed.

SRS document contains all the functional and non-functional requirements along with the use cases that the software must meet.

Software specification requirement document elaborates on the business requirements mentioned in BRD to accommodate the functional and non-functional requirements along with user intersections with the system i.e. use cases.

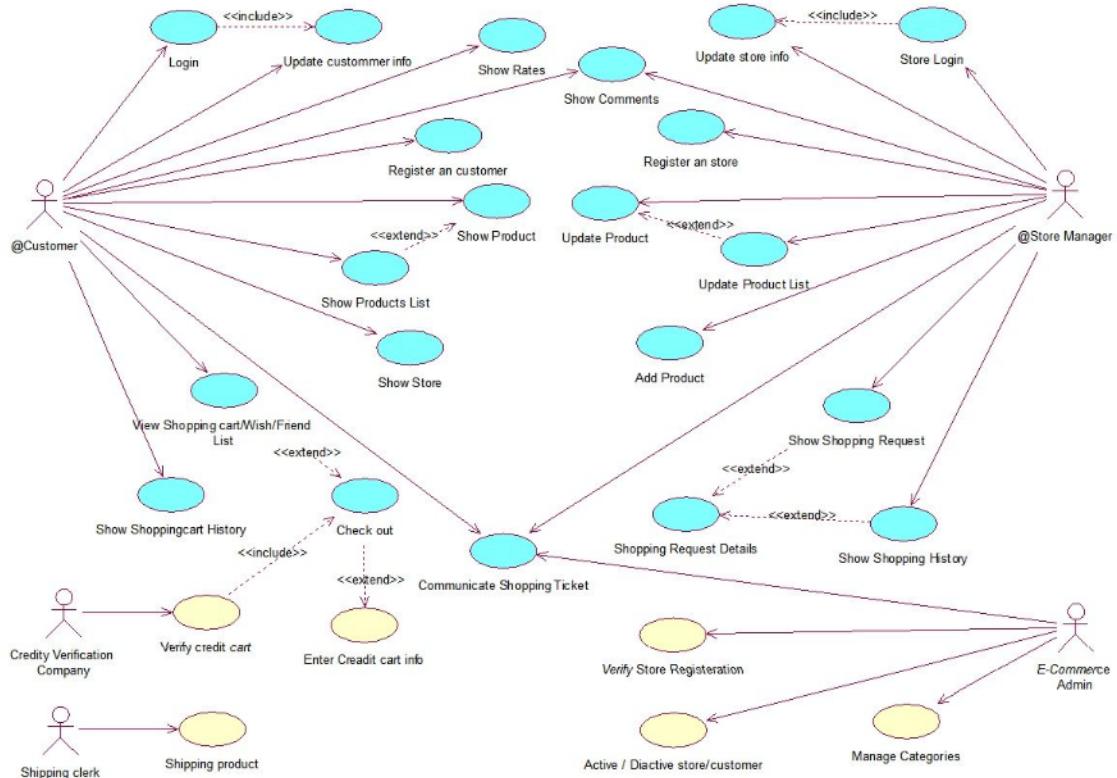
document is prepared by system analysts or business analysts and referred by the project management, development, and implementation teams.



Based on	BRD or BRS	SRS	FRD
What it includes?	BRD or a BRS includes the high-level business requirement of a system to be developed in layman language.	SRS document includes the functional and non-functional requirements and Use Cases.	FRD document consists of detailed requirements in technical terms and technical diagrams like UML, Data Flow, etc.
What is Answers?	BRD answers the <b>WHY</b> part i.e. Why the requirements are being prepared?	SRS answers the <b>WHAT</b> i.e. What requirements to be fulfilled.	FRS focuses on the <b>HOW</b> part i.e. How the requirements will be implemented.
When will it be prepared?	A BRD document is created during the analysis phase of the project.	SRS document is prepared during the planning phase of the project.	FRD or FRS document is also created during the planning phase of the project.
Who will be responsible for creating?	A BRD will be created by the business analysts.	Business Analyst and System Analyst work together to prepare an SRS document.	Since the functional requirement document is detailed and technical, it is created by Business Analyst, System Analysts, and Implementation team together.
Who will be using?	Business Requirements Document is developed for the business users, stakeholders, etc.	SRS document is prepared for the subject matter experts and technical leads.	FRD document will be used by the development team and quality assurance i.e. testing team.

# UML contains different diagrams and Use case is one of it

UML also contains Structural diagrams as well as - such as  
Class Diagram



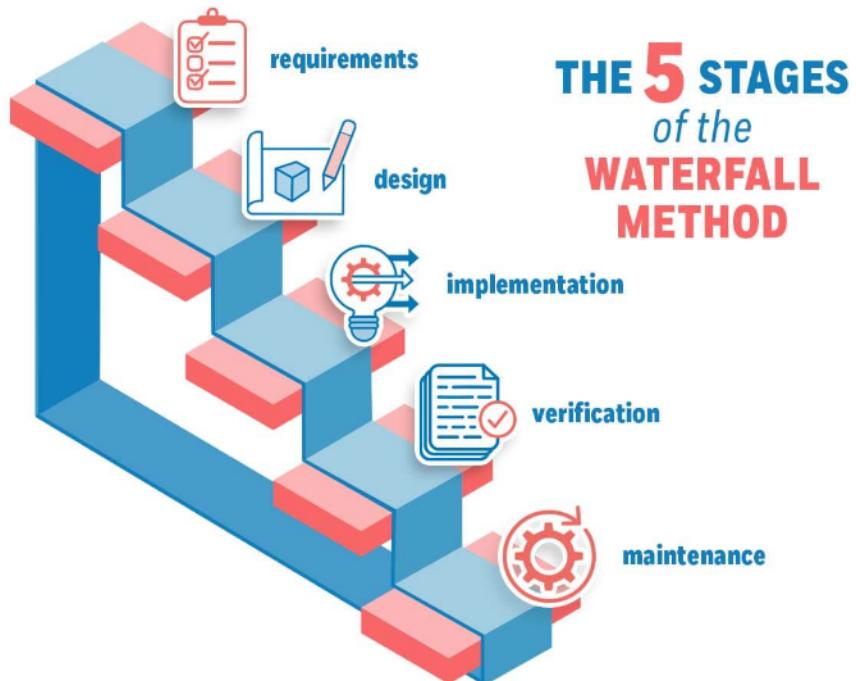
<https://stackoverflow.com/questions/9107448/uml-class-diagram-for-an-e-commerce-website>

# SDLC Models

- Waterfall Model
- Spiral Model
- V-Model
- Agile Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

# Waterfall Model



# Waterfall Model



## ADVANTAGES OF WATERFALL

- ✓ Simple method and easy to use
- ✓ Phases are clear
- ✓ Suitable for smaller projects
- ✓ Easy to manage



## DISADVANTAGES OF WATERFALL

- ✓ Does not allow much revision
- ✓ Not suitable for complex projects
- ✓ Risk and uncertainty are high
- ✓ Does not include a feedback path

## Advantages of Waterfall Model

- The product will be of high quality.
- There are less possibilities of detecting problems because requirement modifications are prohibited.
- Since the testers are employed later, the initial cost is lower.
- Preferred for little projects with frozen criteria.

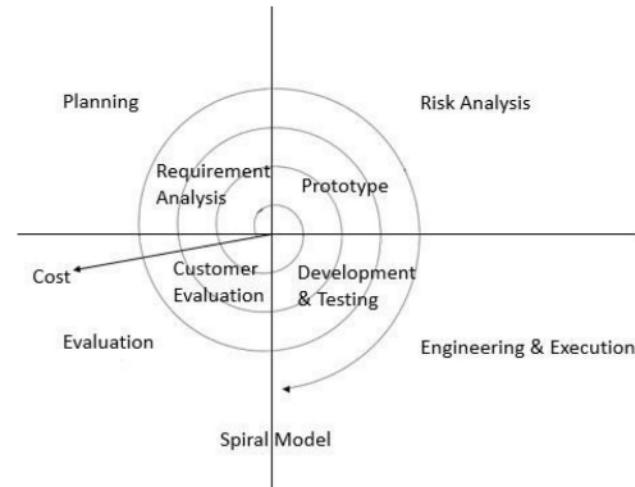
## Disadvantages of Waterfall Model

- Requirement changes are not allowed.
- If there is defect in Requirement that will be continued in later phases.
- Total investment is more because time taking for rework on defect is time consuming which leads to high investment.
- Testing will start only after coding

# Spiral Model

Spiral Model is **iterative model**.

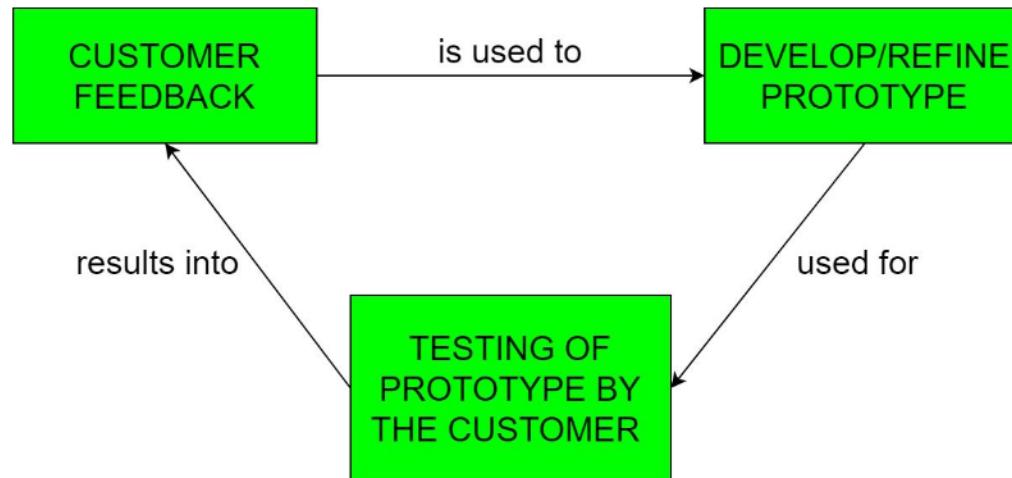
- Spiral Model overcome drawbacks of Waterfall model.
- We follow spiral model whenever there is dependency on the modules.
- In every cycle new software will be released to customer.
- Software will be released in multiple versions. So it is also called version control model.



The Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

# Prototyping Model

Prototyping is defined as the process of developing a working replication of a product or system that has to be engineered. It offers a small scale facsimile of the end product and is used for obtaining customer feedback as described below:



In each phase of the Spiral Model, the features of the product dated and analyzed, and the risks at that point in time are identified and are resolved through prototyping.

Thus, this model is much more flexible compared to other SDLC models.

## Risk Handling in Spiral Model

A risk is any adverse situation that might affect the successful completion of a software project. The most important feature of the spiral model is handling these unknown risks after the project has started. Such risk resolutions are easier done by developing a prototype. The spiral model supports coping up with risks by providing the scope to build a prototype at every phase of the software development.

## Advantages of Spiral Model

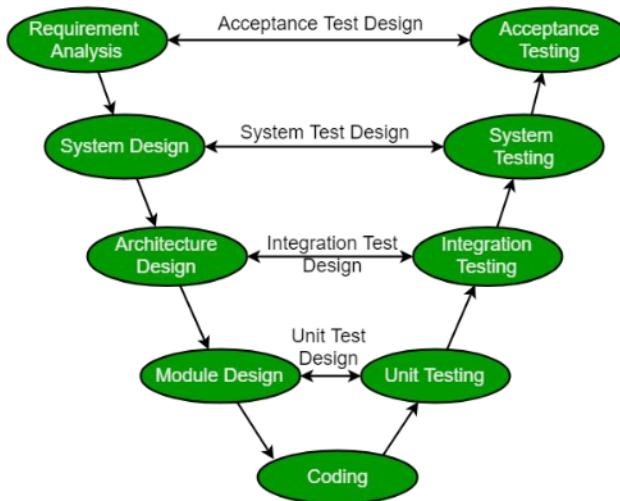
- Testing is done in every cycle, before going to the next cycle.
- Customer will get to use the software for every module.
- Requirement changes are allowed after every cycle before going to the next cycle.

## Disadvantages of Spiral Model

- Requirement changes are NOT allowed in between the cycle.
- Every cycle of spiral model looks like waterfall model.
- There is no testing in requirement & design phase.

# V-Model

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape.



V-Model contains Verification phases on one side of the Validation phases on the other side.

Verification	Validation
Verification means <b>Are we building the software right?</b>	Validation means <b>Are we building the right software ?</b>
Verification is the static testing.	Validation is the dynamic testing.
It does <i>not</i> include the execution of the code.	It includes the execution of the code.
Methods used in verification are reviews, walkthroughs, inspections and desk-checking.	Methods used in validation are Black Box Testing, White Box Testing and non-functional testing.

## VERIFICATION

- 2 sleeves?
- Is it size L?
- Is it blue?
- Are any buttons missing?



## VALIDATION

- Does it fit?
- Is it comfortable to drive in?
- Does the colour match my eyes?
- Can I afford it?
- Is it good quality?
- Will my date like it?

**Verification:** It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.

**Validation:** It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements.

## Advantages

- Testing is involved in each and every phase.

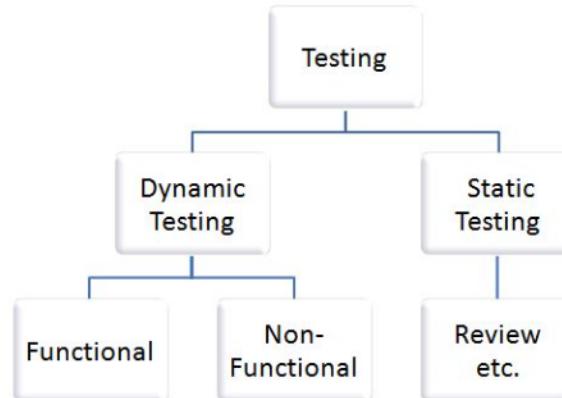
## Disadvantages

- Documentation is more.
- Initial investment is more.

# Static V/S Dynamic Testing

Static testing is an approach **to test project documents in the form of Reviews, Walkthroughs and Inspections.**

Dynamic testing is an approach **to test the actual software by giving inputs and observing results.**



@guru99.com

Static Testing	Dynamic Testing
1. Static Testing is white box testing which is done at early stage of development life cycle. It is more cost effective than dynamic testing.	1. Dynamic Testing on the other hand is done at the later stage of development lifecycle.
2. Static Testing Methods include Walkthroughs, code review.	2. Dynamic testing involves functional and nonfunctional testing
3. It is done before code deployment and without execution of code.	3. It is done after code deployment with the execution of codes.
4. This type of testing comes under Verification.	4. This type of testing comes under Validation.
5. Static testing achieves 100% statement coverage in a relatively short time.	5. Dynamic testing may involve running several test cases, each of which may take longer time.
6. Static testing is about prevention.	6. Dynamic testing is about cure.
7. In Static Testing techniques a checklist is prepared for testing process.	7. In Dynamic Testing technique the test cases are executed.

# AGILE MODEL

The Agile model is a combination of an incremental and iterative approach and is focussed on fitting in well with **flexible requirements**.

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches

the project is divided into small subparts and is delivered in iterations. The subtasks are divided into time frames to serve working functionality with each build. As a result, the final product has all the required features.



Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments.

## Benefits:

Quick development

Quality and measurable results

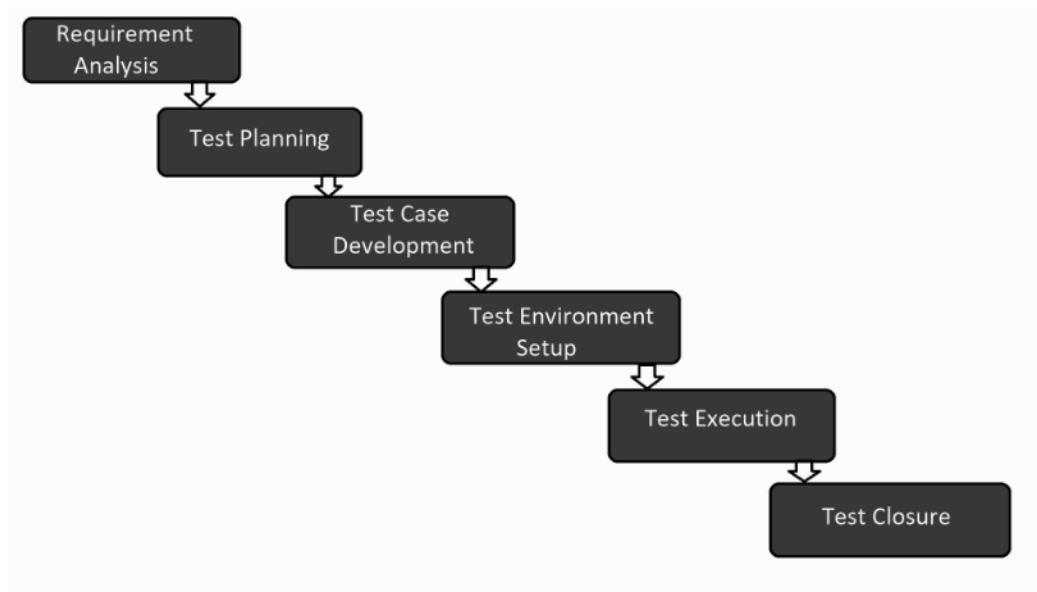
Business value can be delivered –  
demonstrated fast

Requires minimum resources

Highly adaptive to changing requirements

# Software Testing Life Cycle (STLC)

**Software Testing Life Cycle (STLC)** is a sequence of different activities performed during the software testing process.



# Better Resume Tech.

1. **Requirement Analysis:** Quality assurance team understands the requirements like what is to be tested. If anything is missing or not understandable then quality assurance team meets with the stakeholders to better understand the detail knowledge of requirement. - **Documents - SRS, FRD, BRD**
2. **Test Planning:** In this phase manager of the testing team calculates estimated effort and cost for the testing work. This phase gets started once the requirement gathering phase is completed. Documents **Test Plan**

**Test Case Development:** The test case development phase gets started once the test planning phase is completed Test cases. **Documents - Test cases(Excel, GSheet or on Tools)**

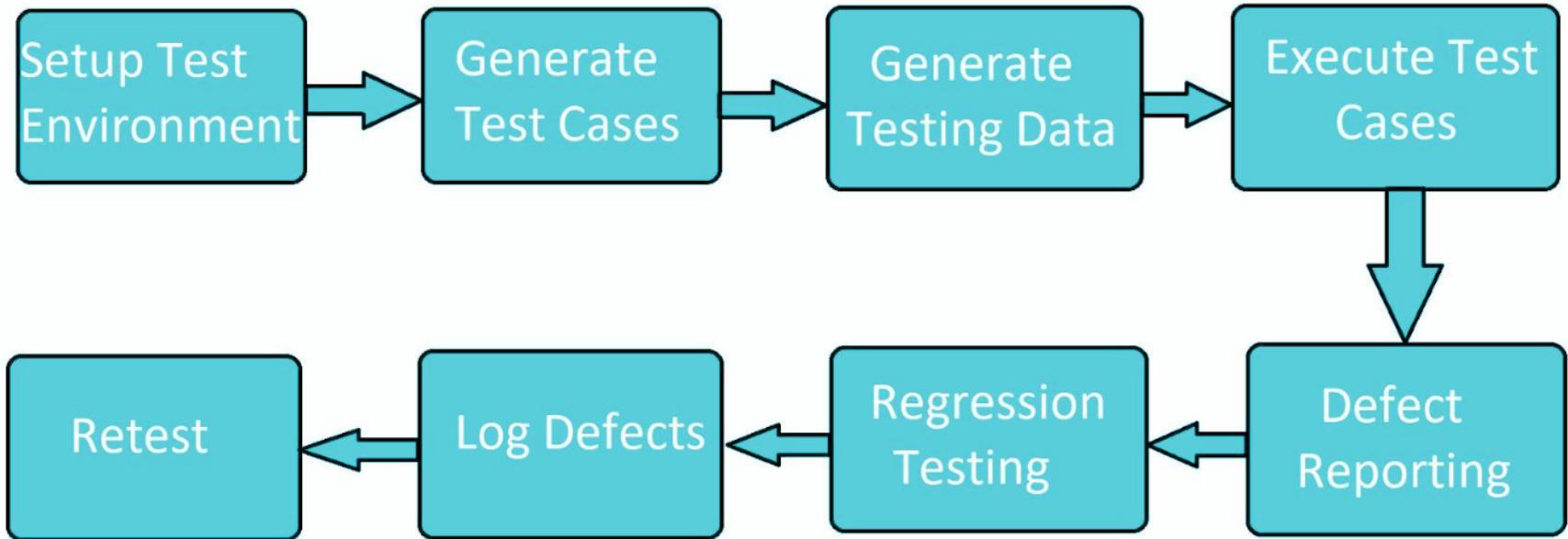
**Test Environment Setup:** Test environment decides the conditions on which software is tested. NA

**Test Execution:** **Documents - Bug Report , Test Execution Report**

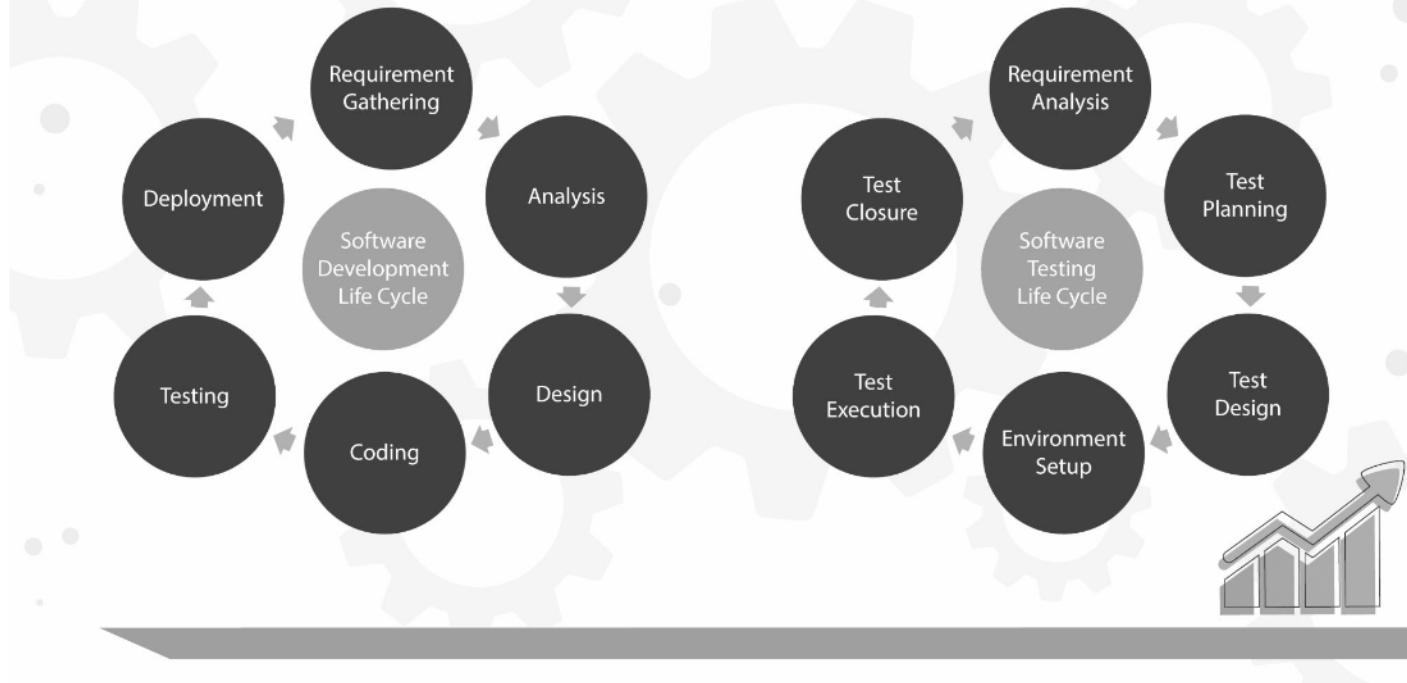
In this phase testing team start executing test cases based on prepared test cases in the earlier step.

**Test Closure: Test Report**

This is the last stage of STLC in which the process of testing is analyzed.



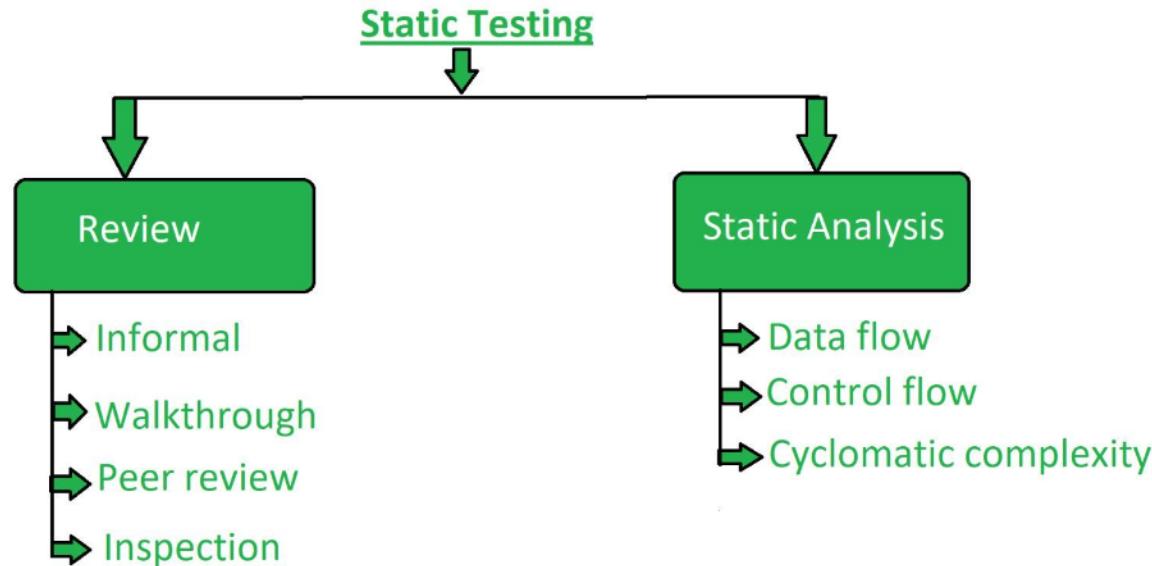
# SDLC vs. STLC



<b>Black Box Testing</b>	<b>White Box Testing</b>
It is also called Specification Based Technique.	It is also called Structural Testing Technique.
Internal structure and coding knowledge is not required.	Internal structure and coding knowledge is required.
Main concentrate on functionality of system.	Main concentrate on code structure ,branches , loops, conditions etc.
Implementation knowledge is not required.	Implementation knowledge is required.

# Static Testing

**Static Testing** is a type of a [Software Testing](#) method which is performed to check the defects in software without actually executing the code of the software application.



# Static Testing done with Static Analysis

Static Analysis:

Static Analysis includes the evaluation of the code quality that is written by developers. Different tools are used to do the analysis of the code and comparison of the same with the standard.

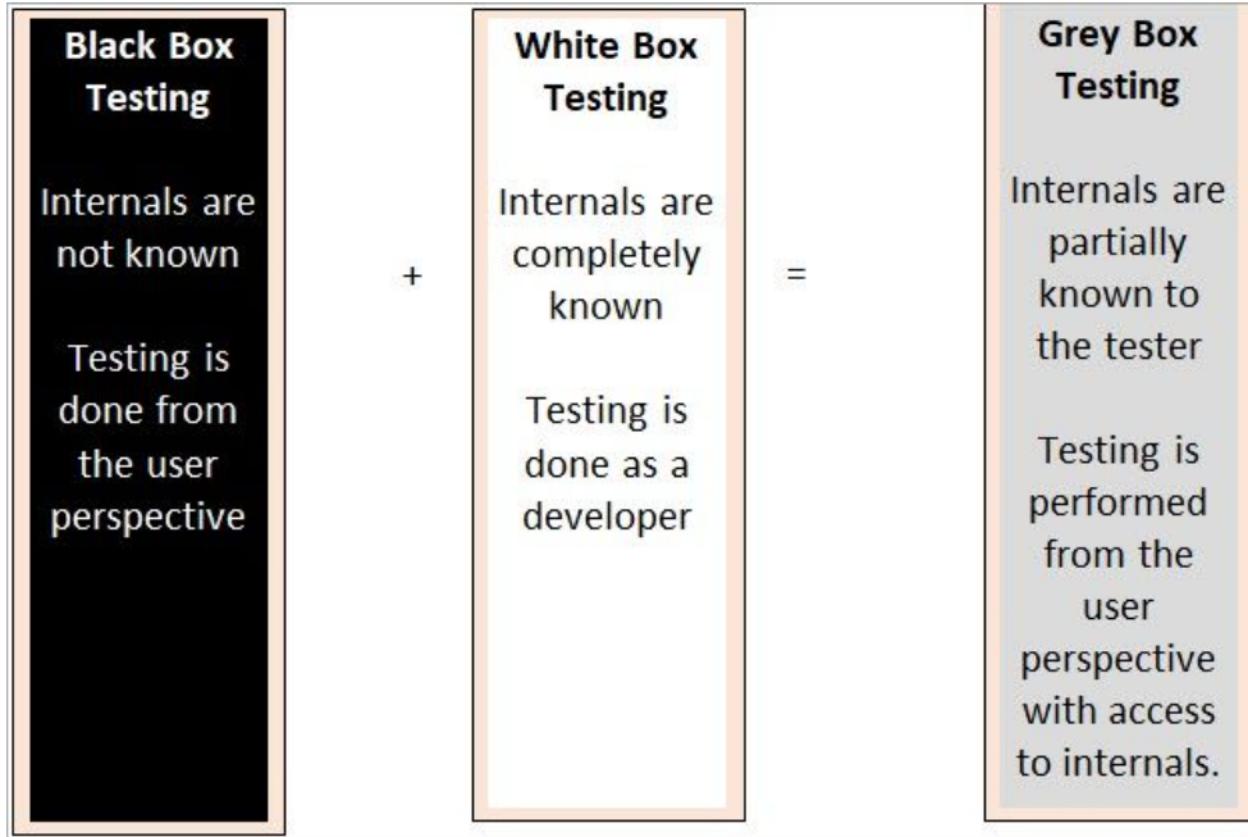
It also helps in following identification of following defects:

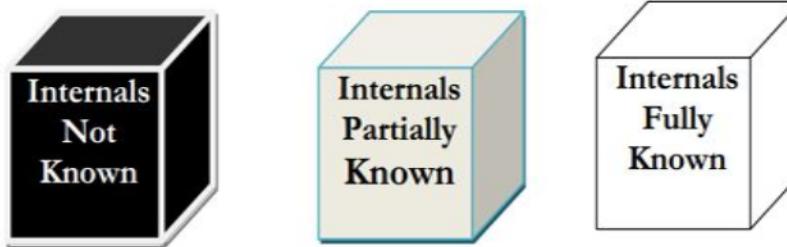
- (a) Unused variables
- (b) Dead code
- (c) Infinite loops
- (d) Variable with undefined value
- (e) Wrong syntax

Static Testing	Dynamic Testing
It is performed in the early stage of the software development.	It is performed at the later stage of the software development.
In static testing whole code is not executed.	In dynamic testing whole code is executed.
Static testing prevents the defects.	Dynamic testing finds and fixes the defects.
Static testing is performed before code deployment.	Dynamic testing is performed after code deployment.
Static testing is less costly.	Dynamic testing is highly costly.
Static Testing involves checklist for testing process.	Dynamic Testing involves test cases for testing process.

# Testing Methodologies

- White box Testing
- Black box Testing
- Grey box Testing

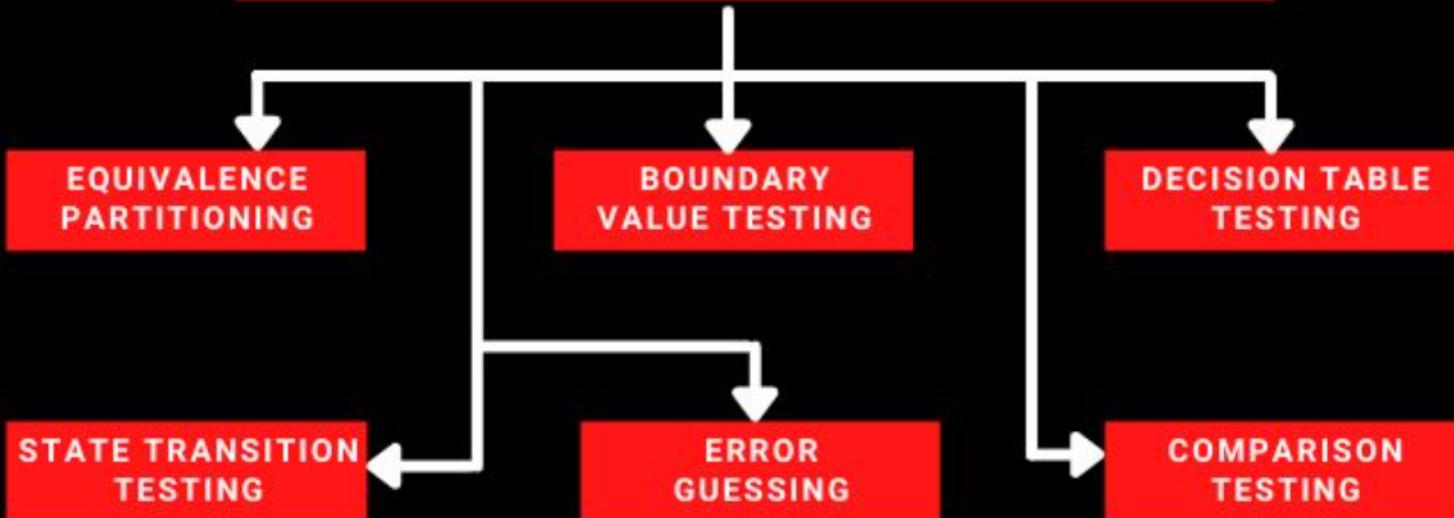




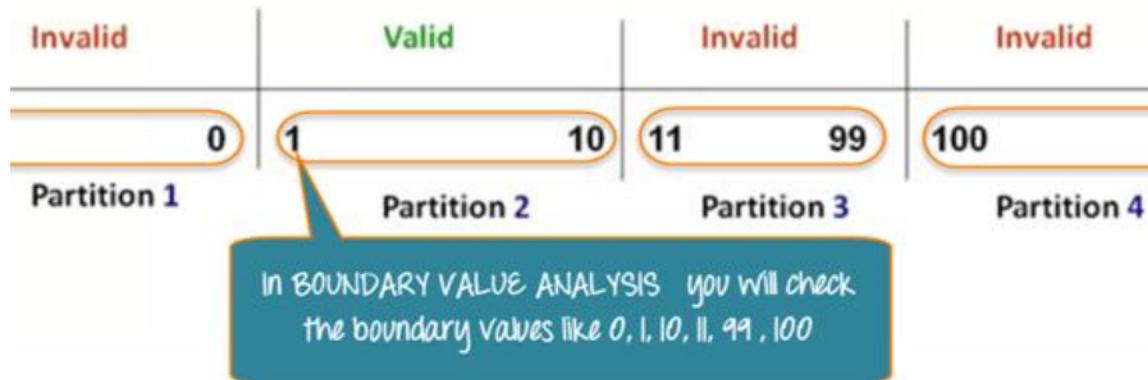
#### Comparison between the Three Testing Types

	<b>Black Box Testing</b>	<b>Grey Box Testing</b>	<b>White Box Testing</b>
1.	The Internal Workings of an application are not required to be known	Somewhat knowledge of the internal workings are known	Tester has full knowledge of the Internal workings of the application
2.	Also known as closed box testing, data driven testing and functional testing	Another term for grey box testing is translucent testing as the tester has limited knowledge of the insides of the application	Also known as clear box testing, structural testing or code based testing
3.	Performed by end users and also by testers and developers	Performed by end users and also by testers and developers	Normally done by testers and developers
4.	-Testing is based on external expectations -Internal behavior of the application is unknown	Testing is done on the basis of high level database diagrams and data flow diagrams	Internal workings are fully known and the tester can design test data accordingly
5.	This is the least time consuming and exhaustive	Partly time consuming and exhaustive	The most exhaustive and time consuming type of testing
6.	Not suited to algorithm testing	Not suited to algorithm testing	Suited for algorithm testing
7.	This can only be done by trial and error method	Data domains and Internal boundaries can be tested, if known	Data domains and Internal boundaries can be better tested

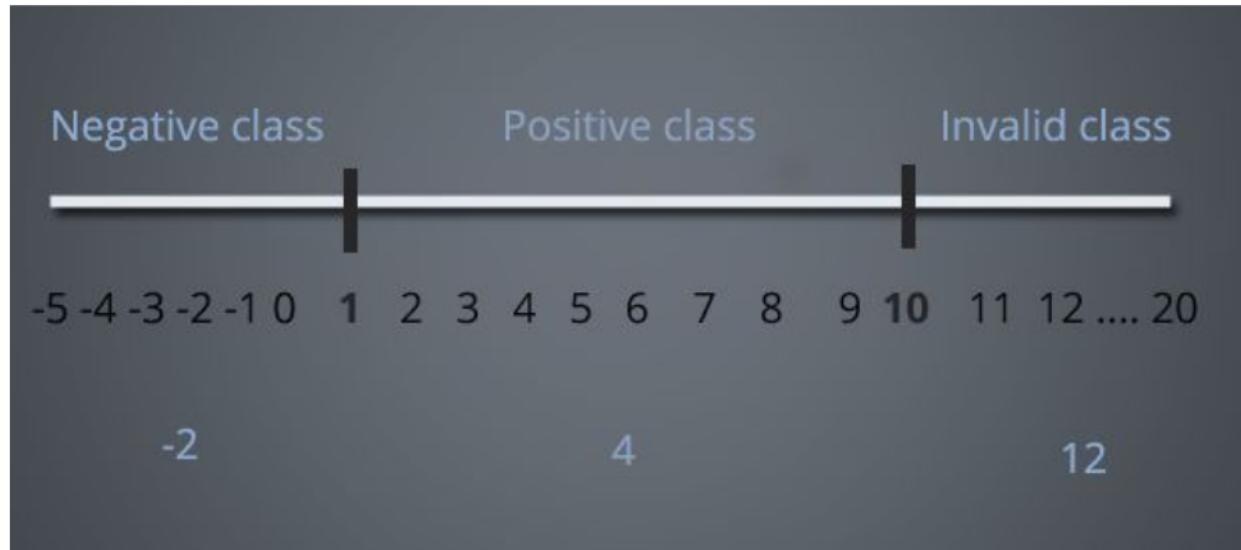
## TECHNIQUES OF BLACK BOX TESTING



# Boundary Value Analysis.



# Equivalence Class Partitioning





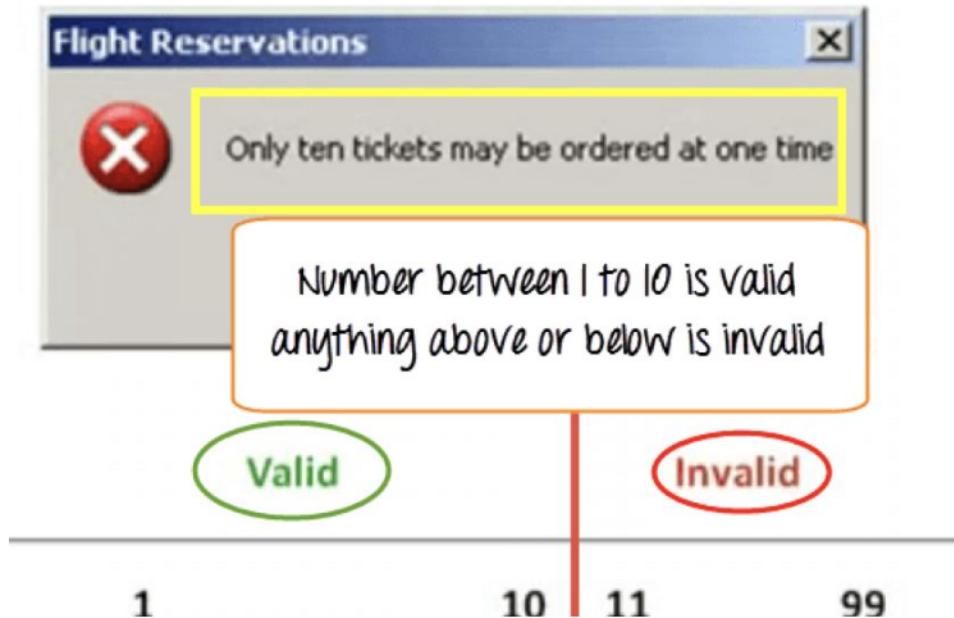
# Practical Test cases

Suppose, In Ecommerce Sale we have discounts like this ,  
Invalid and Valid Test cases

Nil	0- 5000	5000-10000	10000-20000
	10%	15%	25%

<https://forms.gle/hfU6xkKT7jVwmhPR6>

# ECP Vs BVP



Equivalence partitioning and boundary value analysis(BVA) are closely related and can be used together at all levels of testing.

# Problem

Let's consider the behavior of Order Pizza Text Box Below

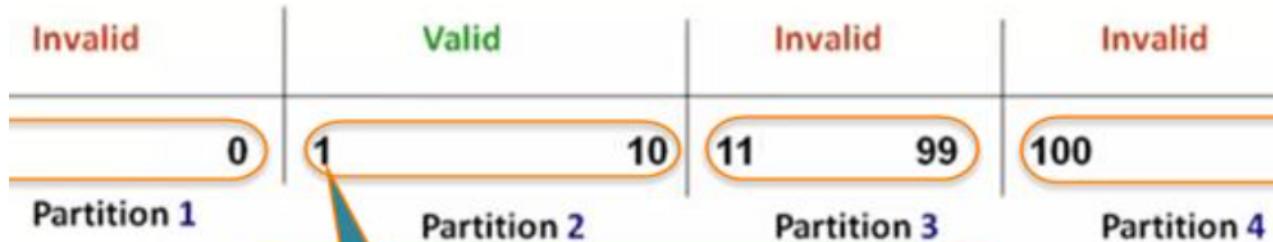
**Pizza values 1 to 10 is considered valid.** A success message is shown.

While value 11 to 99 are considered invalid for order and an error message will appear, “**Only 10 Pizza can be ordered**”

Order Pizza:

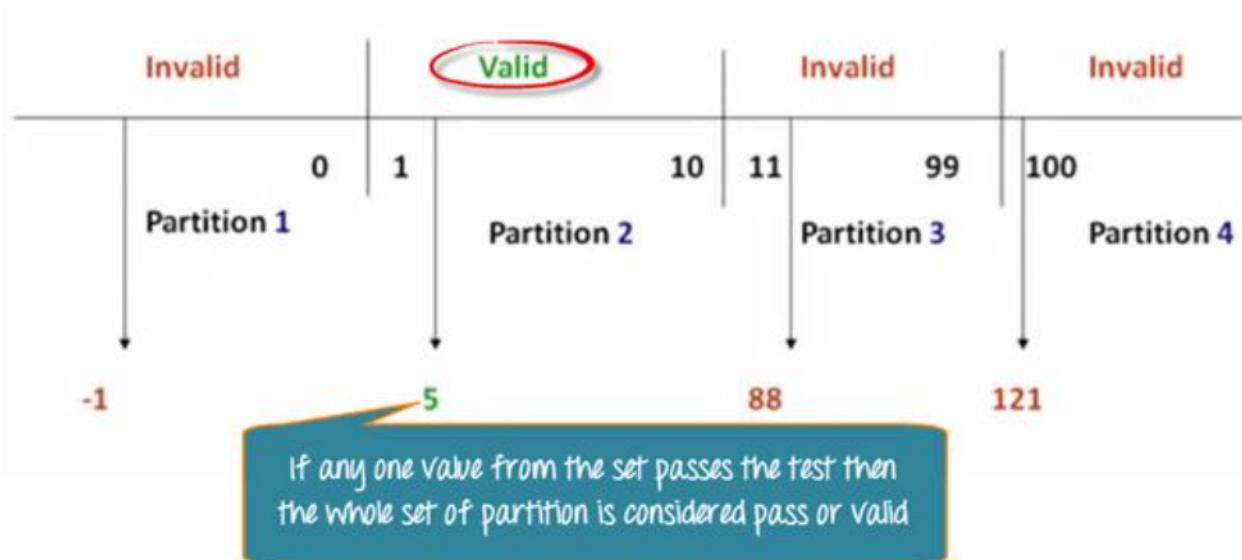
Submit

# Problem



In BOUNDARY VALUE ANALYSIS you will check  
the boundary values like 0, 1, 10, 11, 99, 100

# Problem





# Decision Table based testing

ID	CONDITIONS/ACTIONS	TEST CASE 1	TEST CASE 2	TEST CASE 3	TEST CASE 4
Condition 1	Valid User ID	T	T	F	F
Condition 2	Valid Password	T	F	T	F
Action 1	Home Page	Execute			
Action 2	Show a Message as 'Invalid User Credentials'		Execute	Execute	Execute

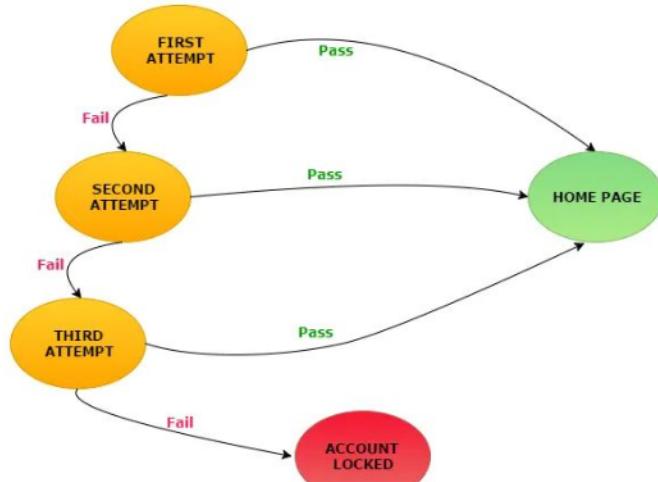


# Error Guessing

Intuition	Experience
<p>The program reads a file. What happens if the program gets an empty file or the file does not exists?</p> <p>Tester would create test cases for those conditions.</p>	

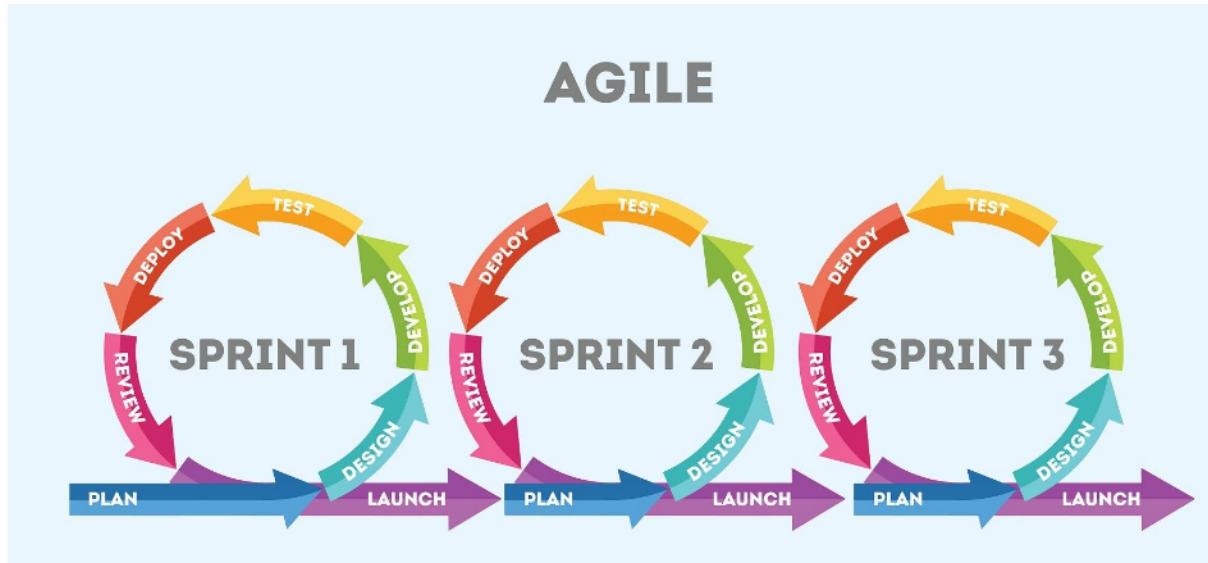
# State Transition

Take an example of login page of an application which locks the user name after three wrong attempts of password.



# Agile

Agile is an **iterative approach** to project management and **software development** that helps teams deliver value to their customers faster and with fewer headaches





## Agile Manifesto

# Agile Values

We are uncovering **better ways of developing software by doing it and helping others do it**. Through this work we have come to value:



Individuals and interactions *over* processes and tools



Working software *over* comprehensive documentation



Customer collaboration *over* contract negotiation



Responding to change *over* Following a plan

# Agile Manifesto

# 12 agile principles

---

## in software development



Customer  
satisfactions



Changing  
requirements



Frequent  
delivery



Communicate  
regularly



Support  
team member



Face-to-face  
communication



Measure  
work progress



Development  
process



Good  
design



Measure  
progress



Continue  
seeking result



Reflect and  
adjust regularly



# Agile

## The Agile Scrum Framework at a glance

