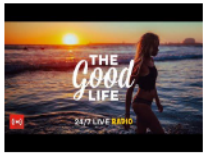
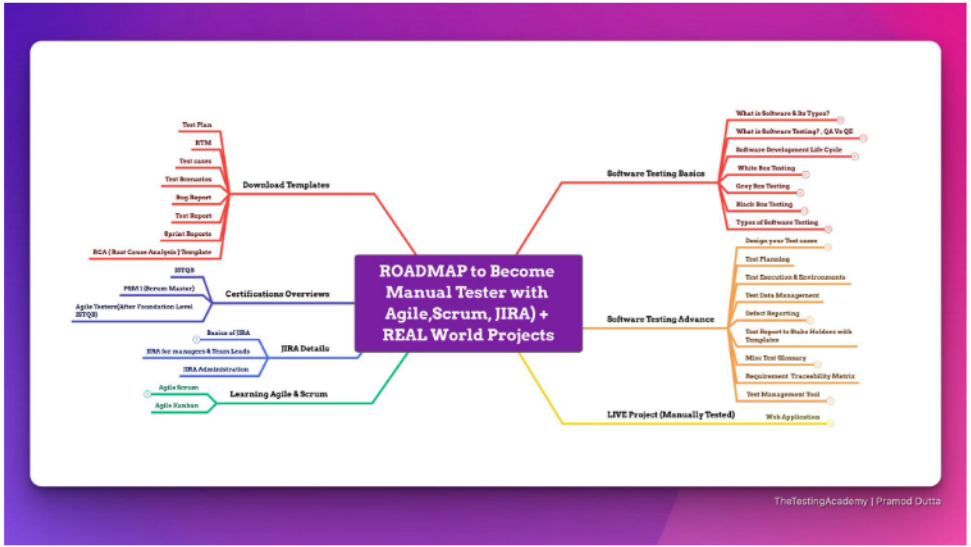
Meeting Etiquette

- 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



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

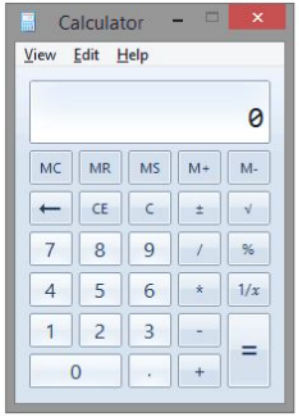
New Action

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

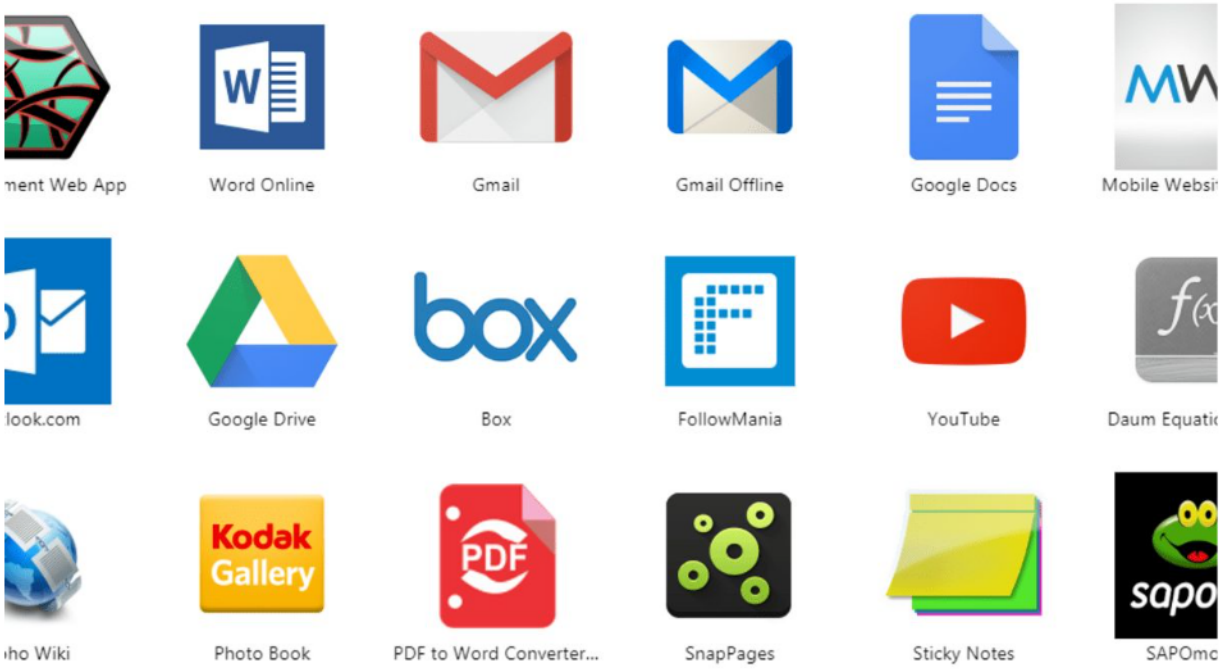
What is Software?

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

Windows Calculator



What is Software?



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

end result meets the business and user requirements.

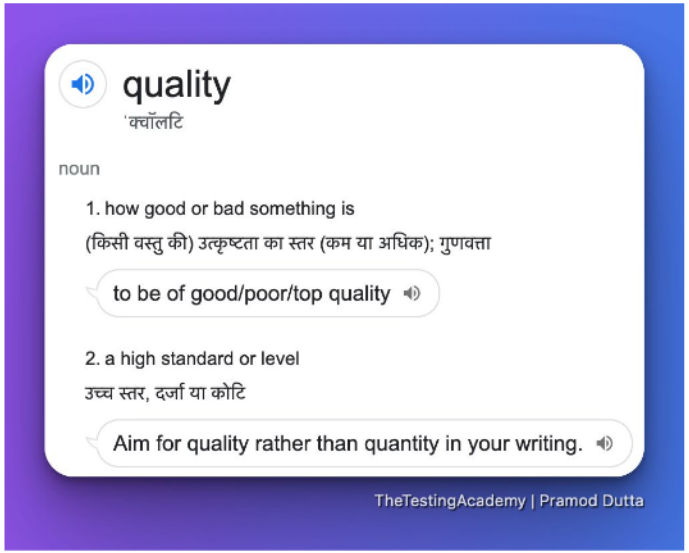
ensure that it satisfies the BRS that is Business Requirement

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


Prevent Defects
1
Evaluate Work Product
2
Verify Requirements
3
Validate Test Objects
4
Build Confidence
5
Reduce Risk
6
Share Information To Stakeholders
7
Find Failure & Defects
8

Quality

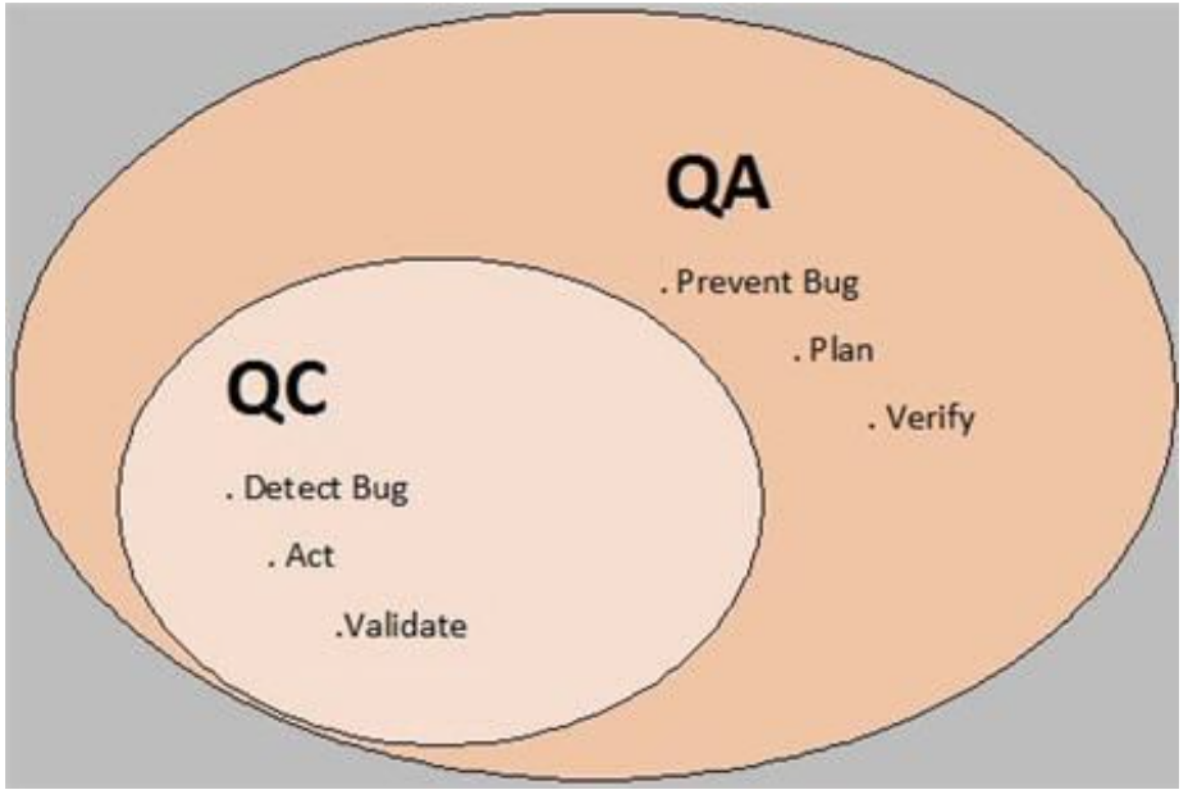
Quality is defined as justification of all the requirements of a customer in a product.



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.	


QA
. Prevent Bug
. Plan
. Verify
QC
. Detect Bug
. Act
.Validate

Quality Assurance vs Testing

Comparison between QA, QC and Testing

Quality Assurance

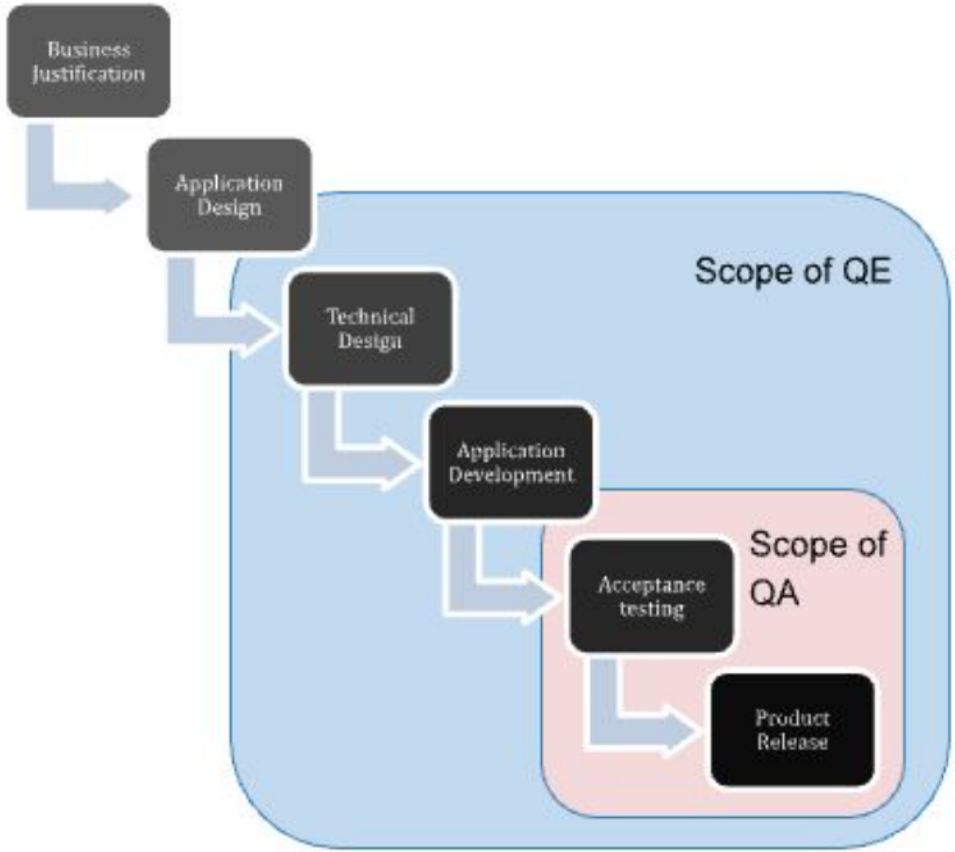
- Subset of SDLC
- Process oriented
- Ensure that processes and procedures are in place to achieve quality
- Focus on process to achieve required quality
- Prevent defects
- Whole team approach
- Proactive process

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 orinted
- Validate the product against specifications
- Focus on actual testing of the product
- Find and fix defects
- Reactive process
- Testing team


Business Justification
Application Design
Technical Design
Application Development
Acceptance testing
Product Release
Scope of QE
Scope of QA

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?

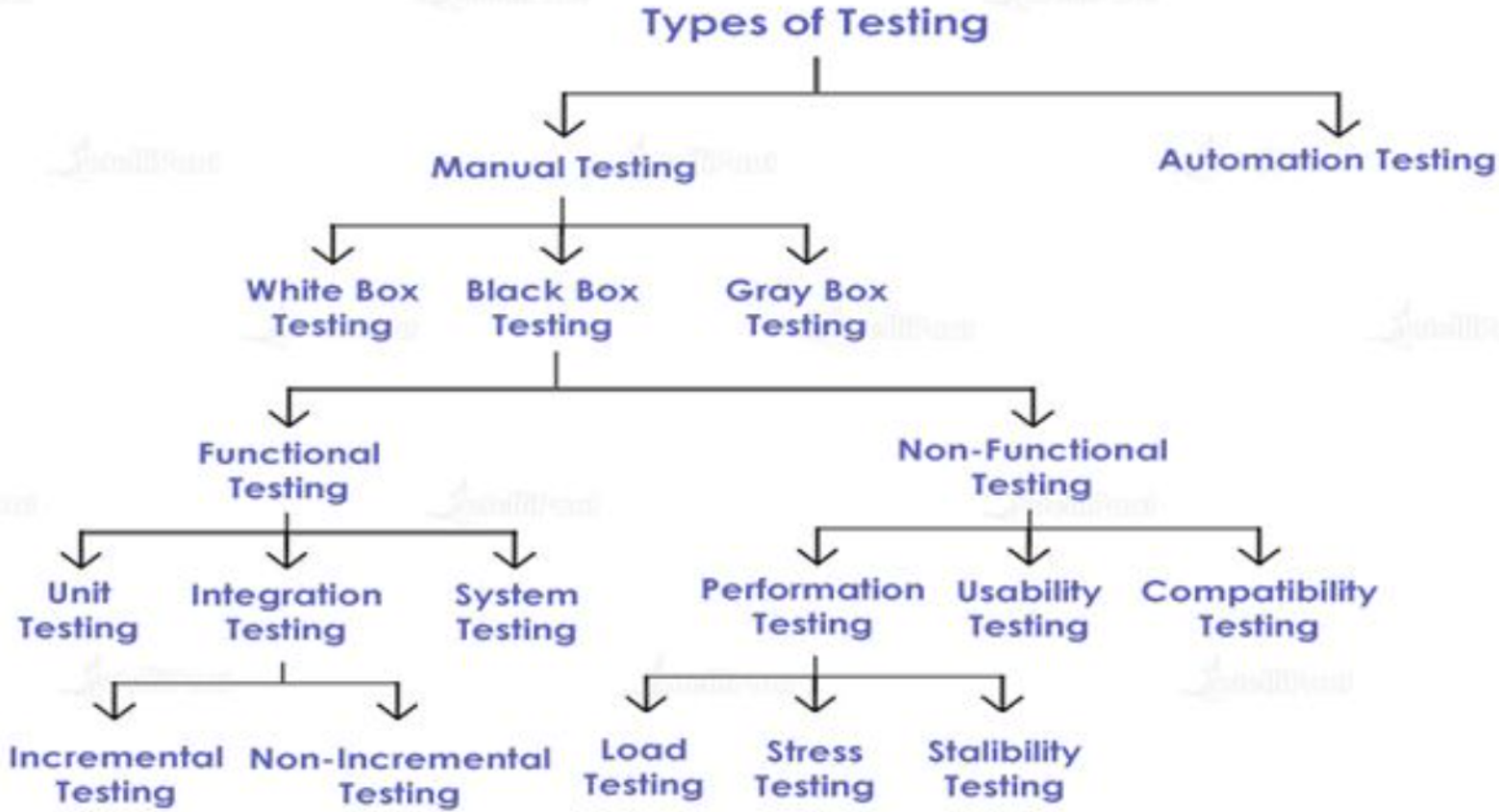
Vulnerability in Windows 10. This bug enables users to escape from security sandboxes through a flaw in the win32k system

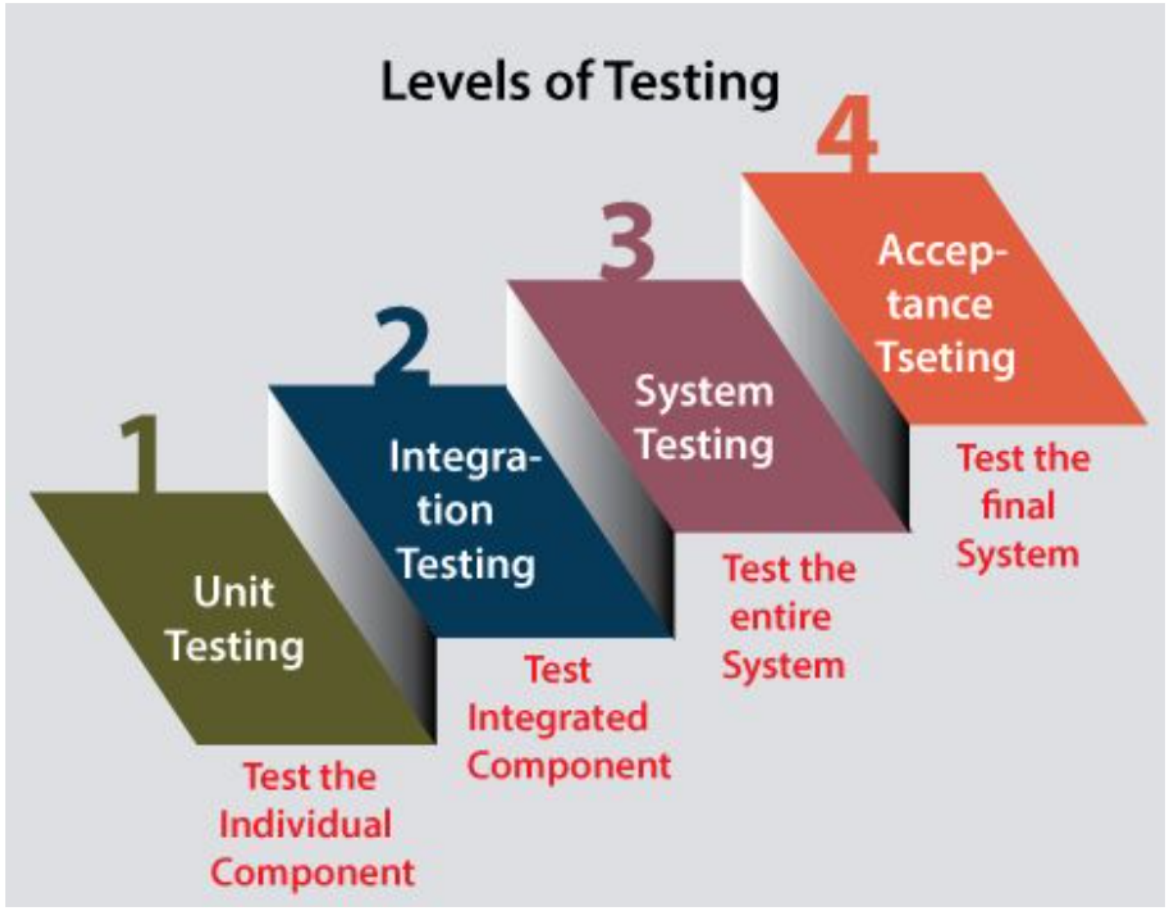
China Airlines Airbus A300 crashed due to a software bug on April 26, 1994, killing 264 innocents live

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

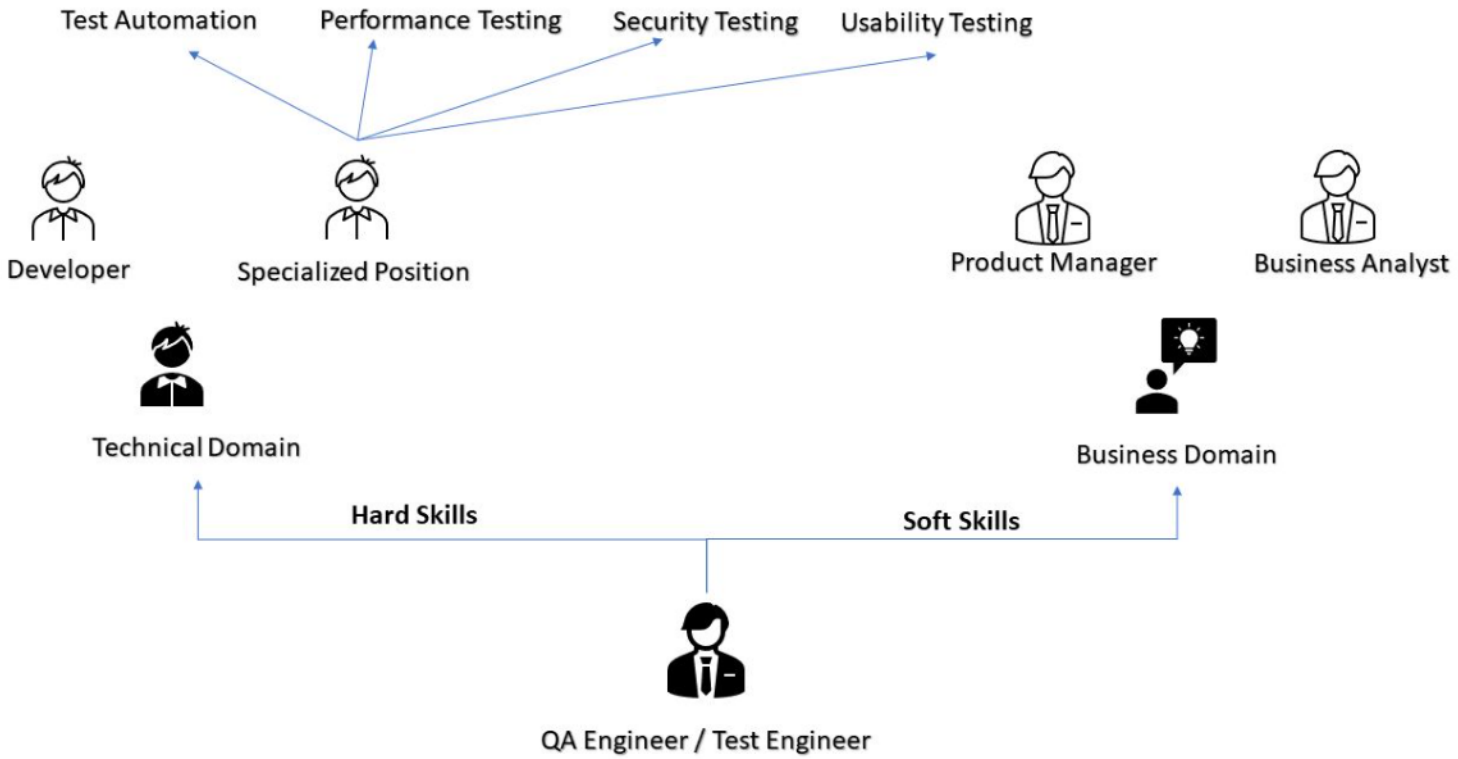
Benefits of Software Testing

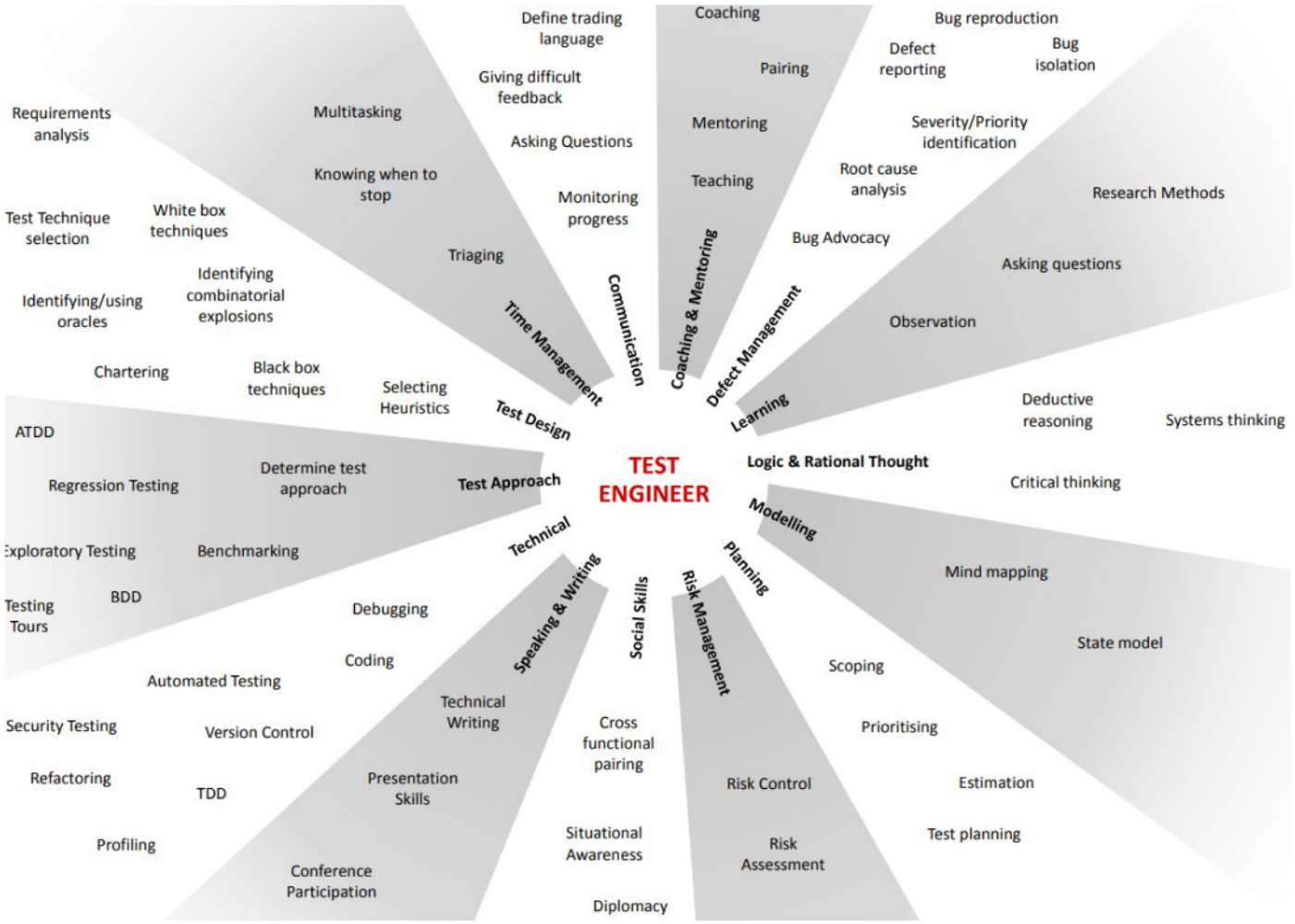
- Cost-Effective
- Product quality
- **Customer Satisfaction**


Types of Testing
Manual Testing
Automation Testing
White Box Testing
Black Box Testing
Gray Box Testing
Functional Testing
Non-Functional Testing
Unit Testing
Integration Testing
System Testing
Performation Testing
Usability Testing
Compatibility Testing
Incremental Testing
Non-Incremental Testing
Load Testing
Stress Testing
Stalibility Testing


Levels of Testing
1
Unit Testing
Test the Individual Component
2
Integra-tion Testing
Test Integrated Component
3
System Testing
Test the entire System
4
Accep-tance Tseting
Test the final System


Trainee Tester
Software Tester
QA Analyst
Software Test Engineer
QA Team Coordinator
Test Manager
Senior Test Manager


Test Automation
Performance Testing
Security Testing
Usability Testing
Developer
Specialized Position
Product Manager
Business Analyst
Technical Domain
Business Domain
Hard Skills
Soft Skills
QA Engineer / Test Engineer

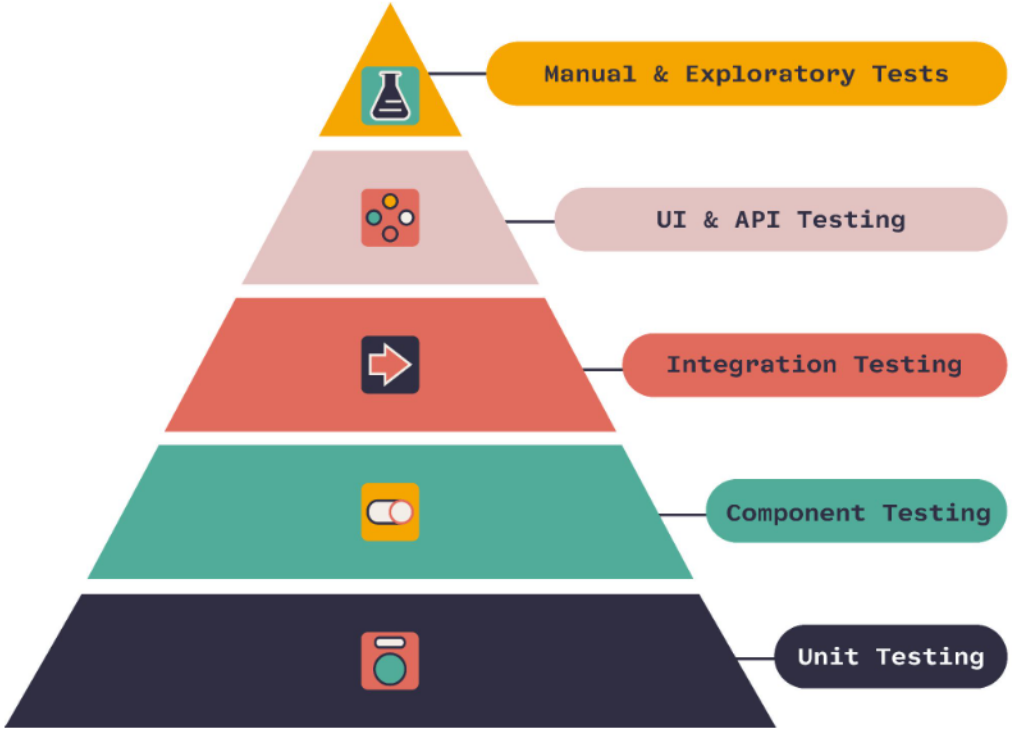

TEST ENGINEER
Time Management
Communication
Coaching & Mentoring
Defect Management
Learning
Test Design
Test Approach
Technical
Speaking & Writing
Social Skills
Risk Management
Planning
Modelling
Logic & Rational Thought
Define trading language
Giving difficult feedback
Asking Questions
Monitoring progress
Triaging
Multitasking
Knowing when to stop
Requirements analysis
White box techniques
Test Technique selection
Identifying combinatorial explosions
Identifying/using oracles
Chartering
Black box techniques
Selecting Heuristics
Coaching
Pairing
Mentoring
Teaching
Bug Advocacy
Bug reproduction
Defect reporting
Bug isolation
Severity/Priority identification
Root cause analysis
Research Methods
Asking questions
Observation
Deductive reasoning
Systems thinking
Critical thinking
ATDD
Regression Testing
Determine test approach
Exploratory Testing
Benchmarking
Testing Tours
BDD
Debugging
Coding
Automated Testing
Security Testing
Version Control
Refactoring
TDD
Profiling
Technical Writing
Presentation Skills
Conference Participation
Cross functional pairing
Situational Awareness
Diplomacy
Risk Control
Risk Assessment
Scoping
Prioritising
Estimation
Test planning
Mind mapping
State model

Types of Software Testing

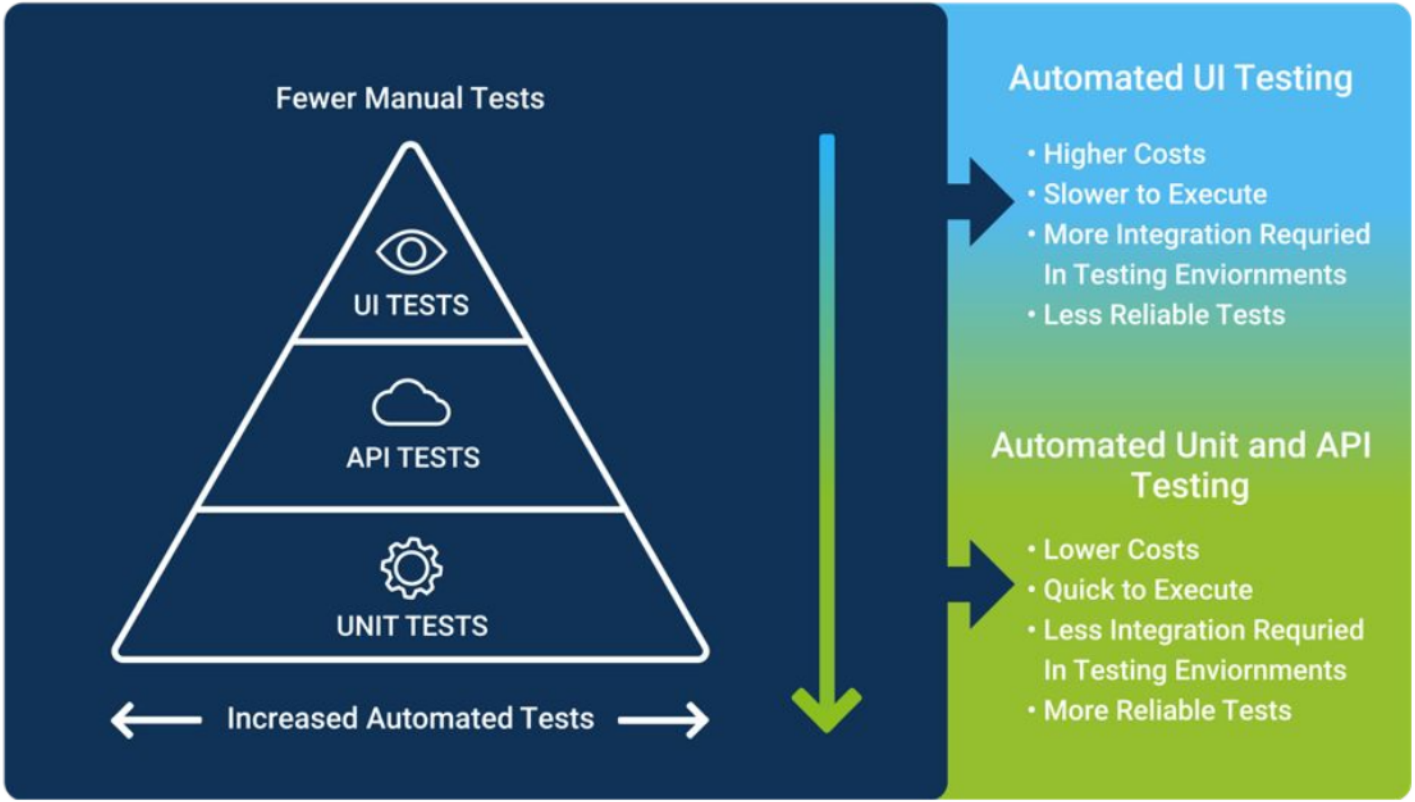
Functional Testing

Non-Functional Testing

Testing Pyramid



The Automation Pyramid



Unit Testing

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

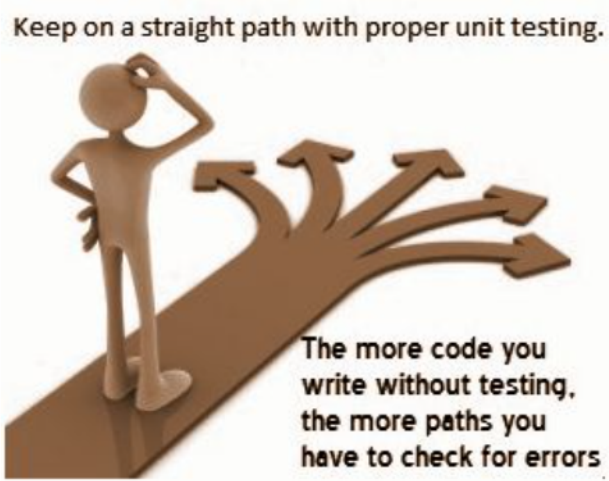
Unit tests help to fix bugs early in the development cycle and save costs.

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



Unit Testing

Demo of Web App unit

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

isSumPositive function

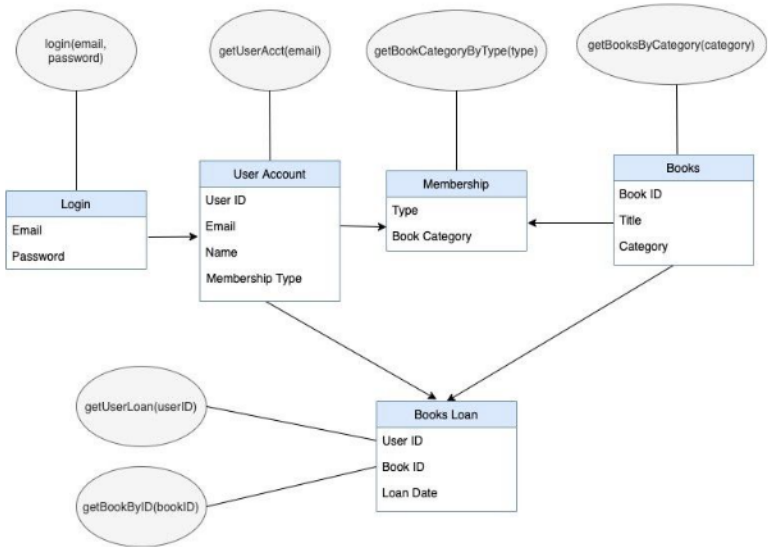
```
@Test
public void shouldAnswerWithPass()
{
    Assert.assertTrue(AddModule.isSumPositive( a: 2, b: 3));
}

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

Integration Testing

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

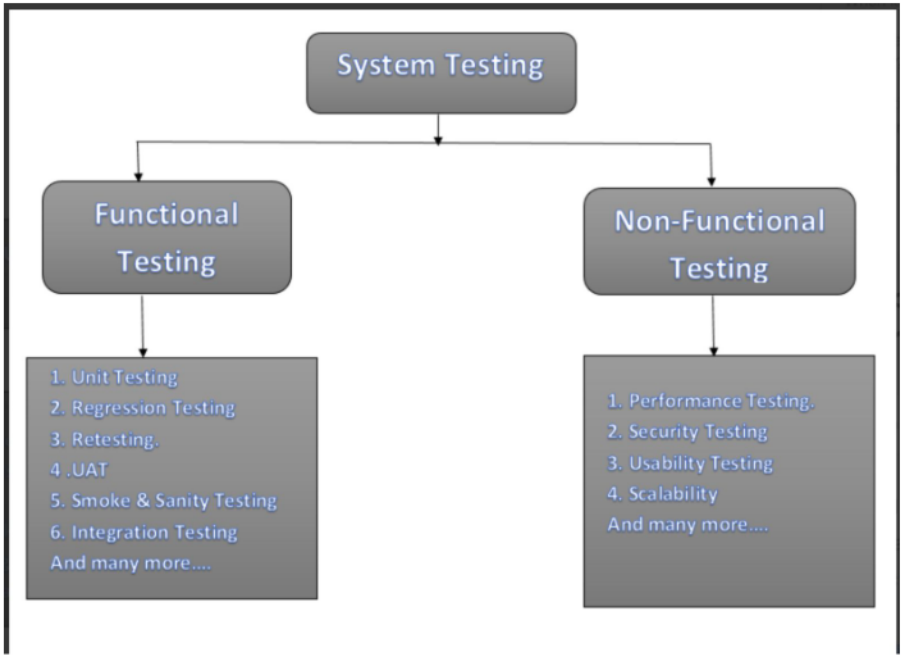
Software modules are integrated logically and tested as a group.



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	End-to-end Testing
It is carried out once integration testing is performed.	It is performed after the system testing.

Levels of Testing

UNIT TESTING
Test Individual Component

INTEGRATION TESTING
Test Integrated Component

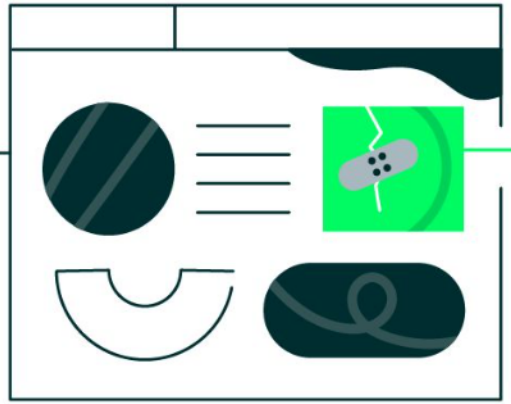
SYSTEM TESTING
Test the entire System

ACCEPTANCE TESTING
Test the final System

Regression testing vs. retesting

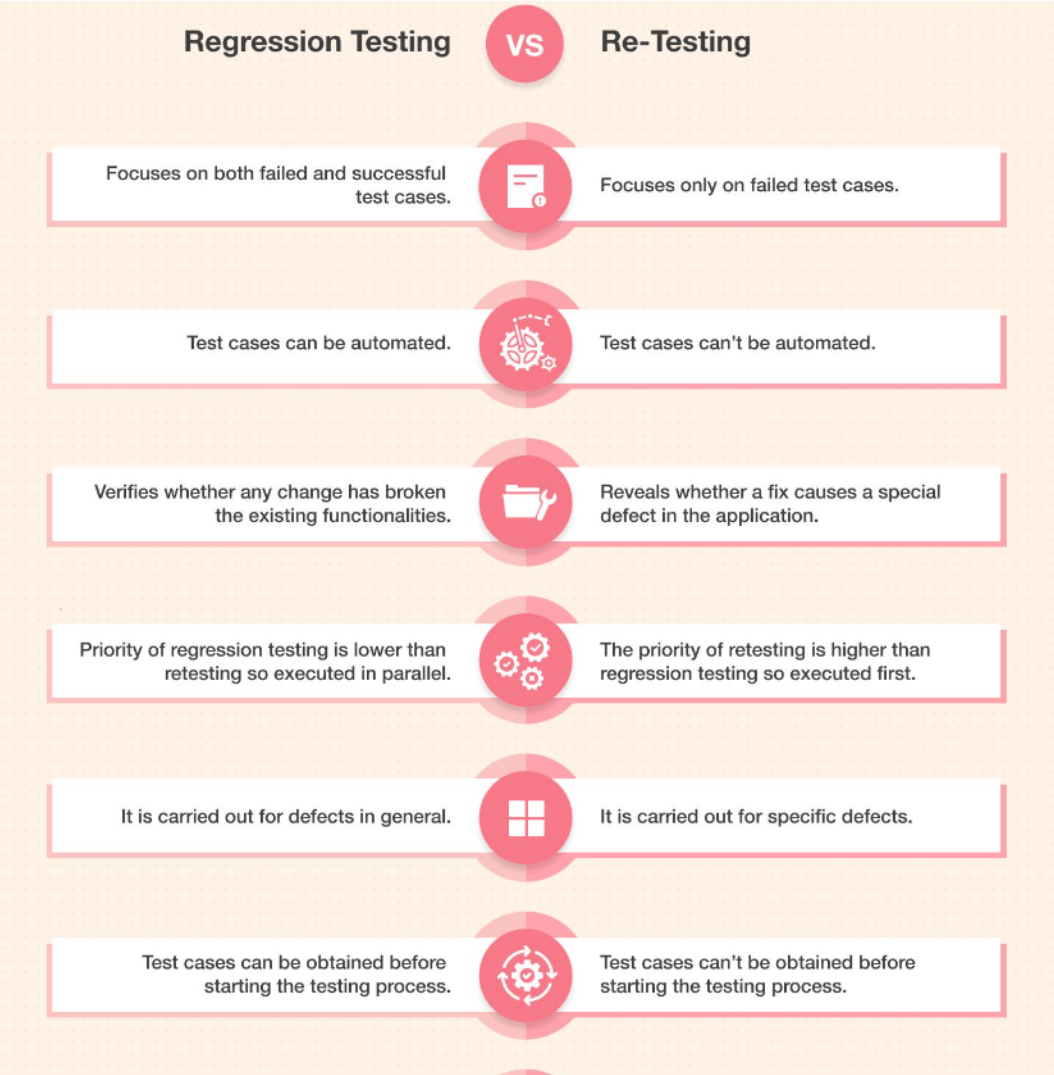
Regression testing

Testing code which was working, but now, due to updates, might not be



Retesting

Testing code which was not working, but which you believe to have been fixed



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

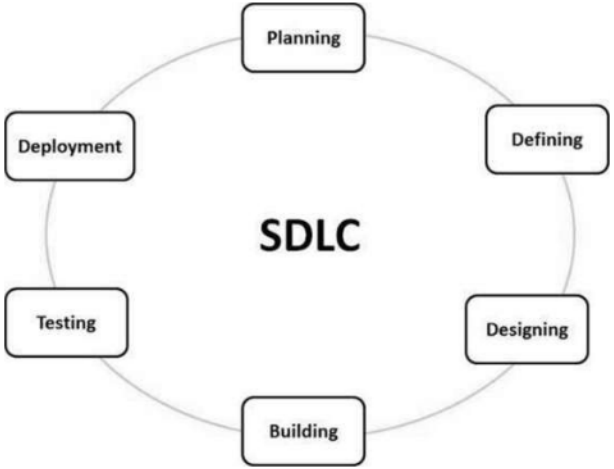
What you do can extra? As Manual tester

- 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

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.



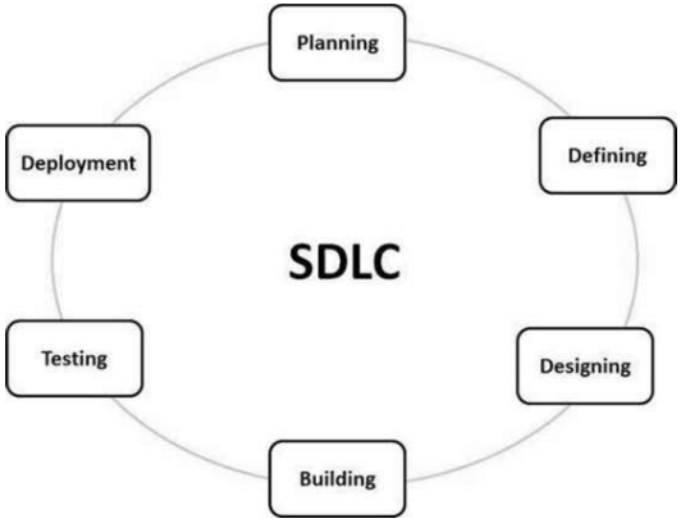
1. Planning and Requirement Analysis

- 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/EeXpfJ0W



2. Defining Requirements

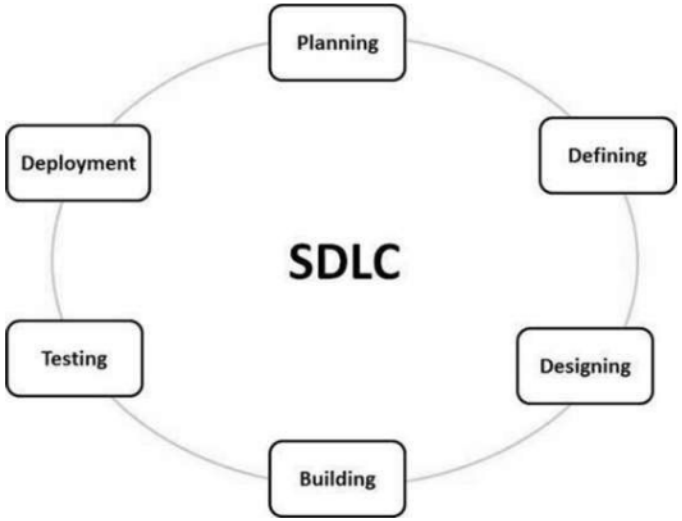
- Define and document the product requirements and get them approved from the customer or the market analysts

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/



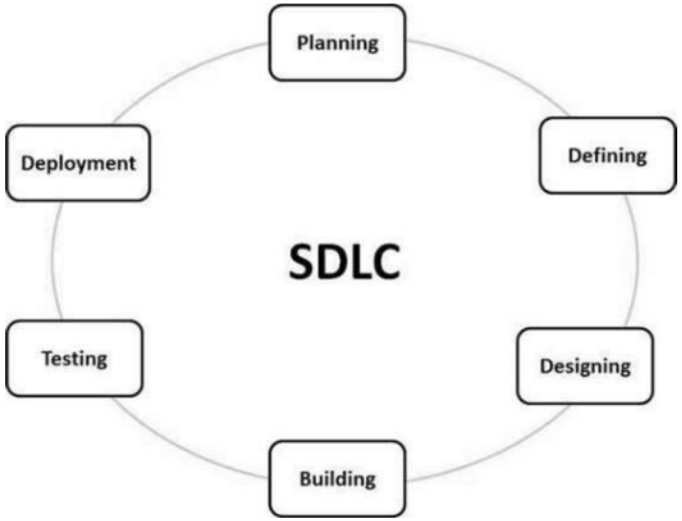
3. Designing

- 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



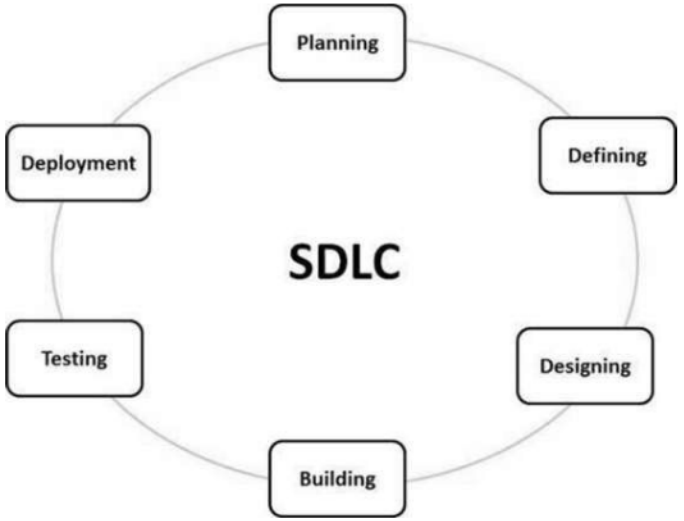
4. Building

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

Documents - DDS

https://sdet.live/samplesrs

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



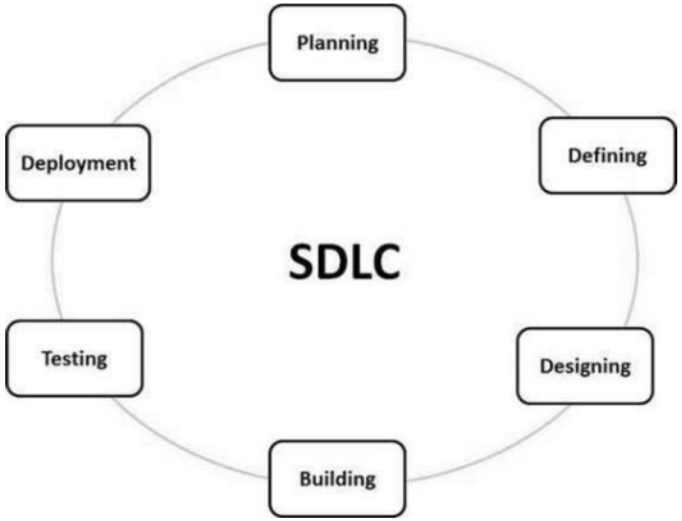
5. Testing

- 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

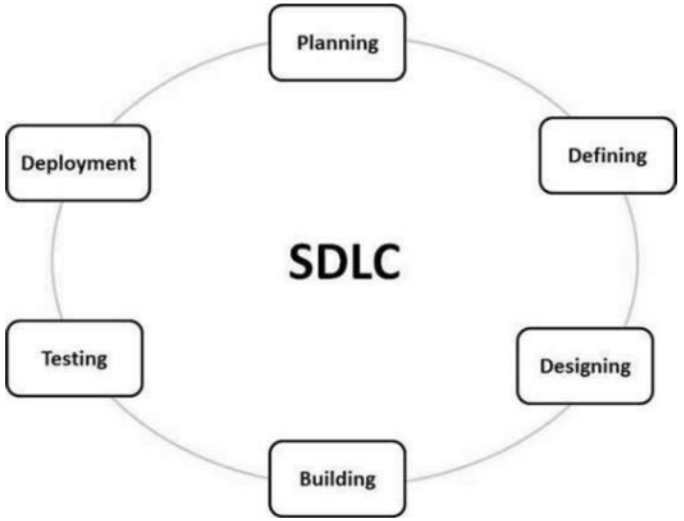


6. Deployment

- 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).

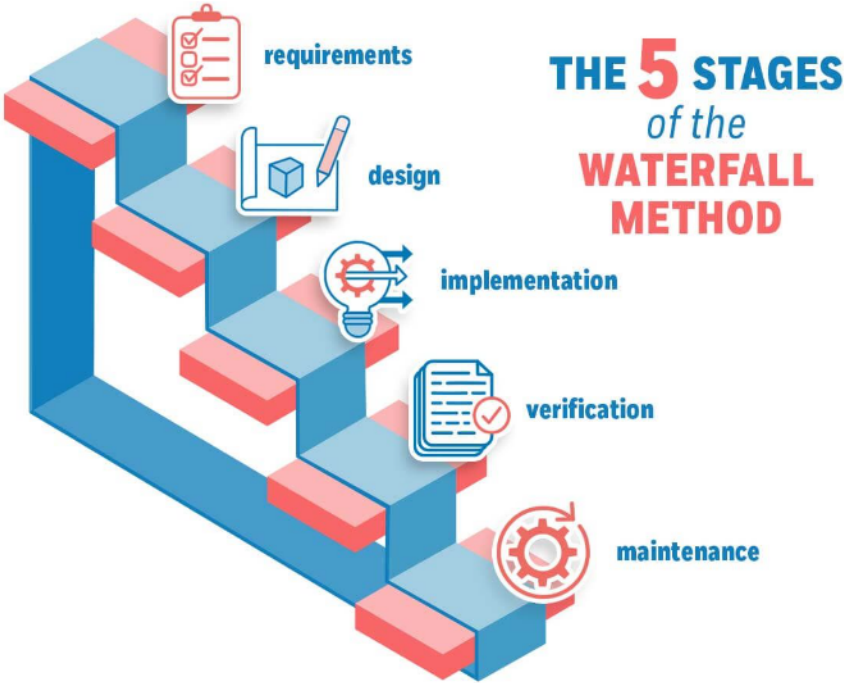


SDLC Models

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang 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

Iterative Model

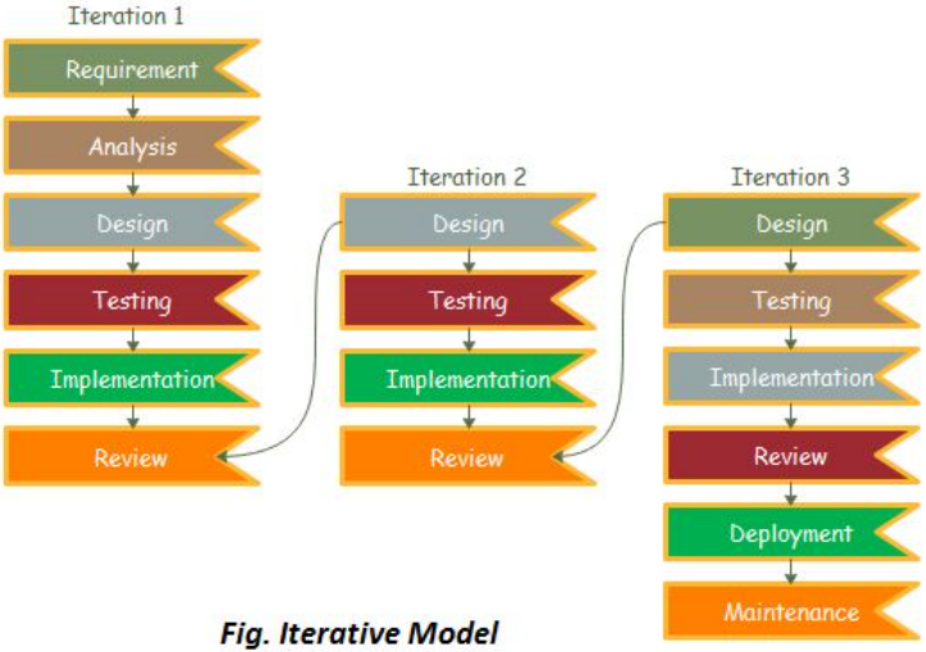
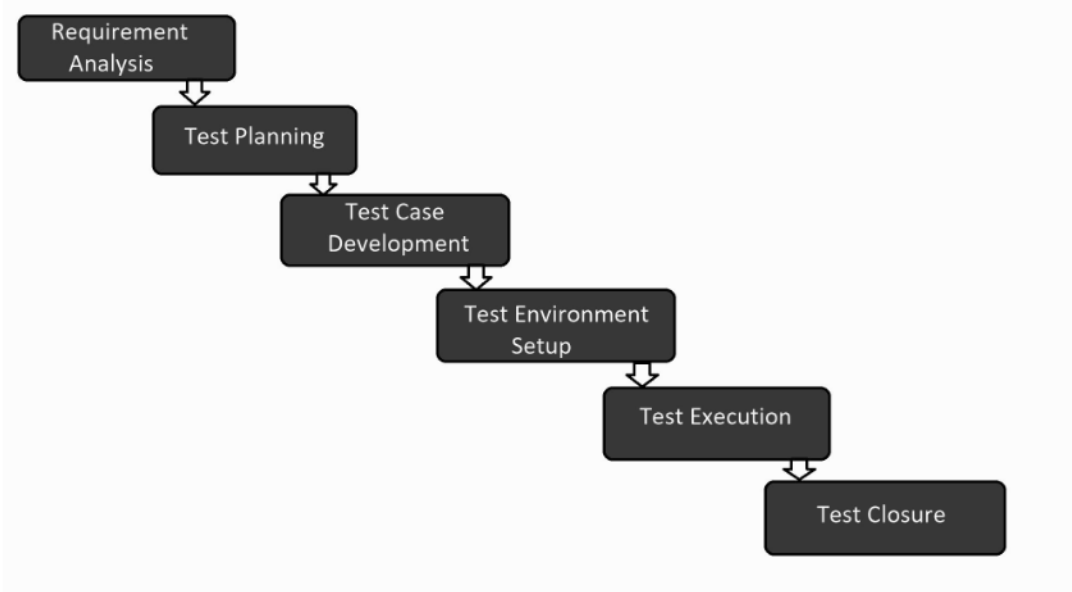


Fig. Iterative Model

Software Testing Life Cycle (STLC)

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



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)**

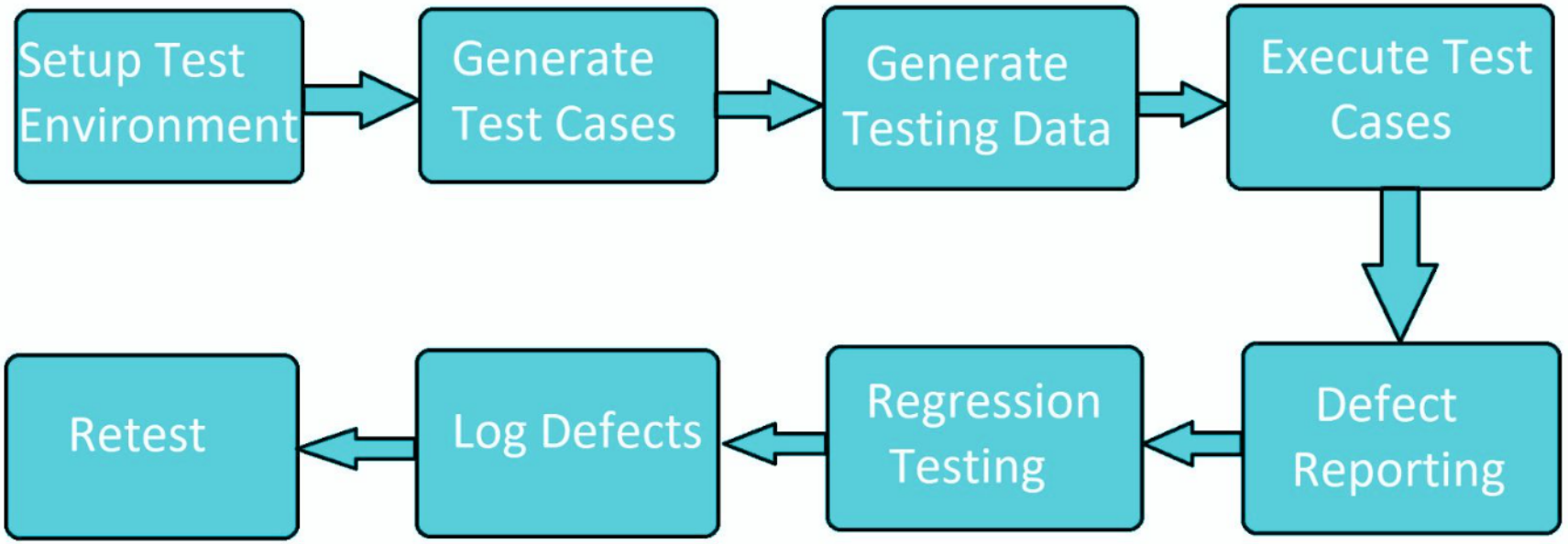
Test Environment Setup: Test environment decides the conditions on which software is tested. NA

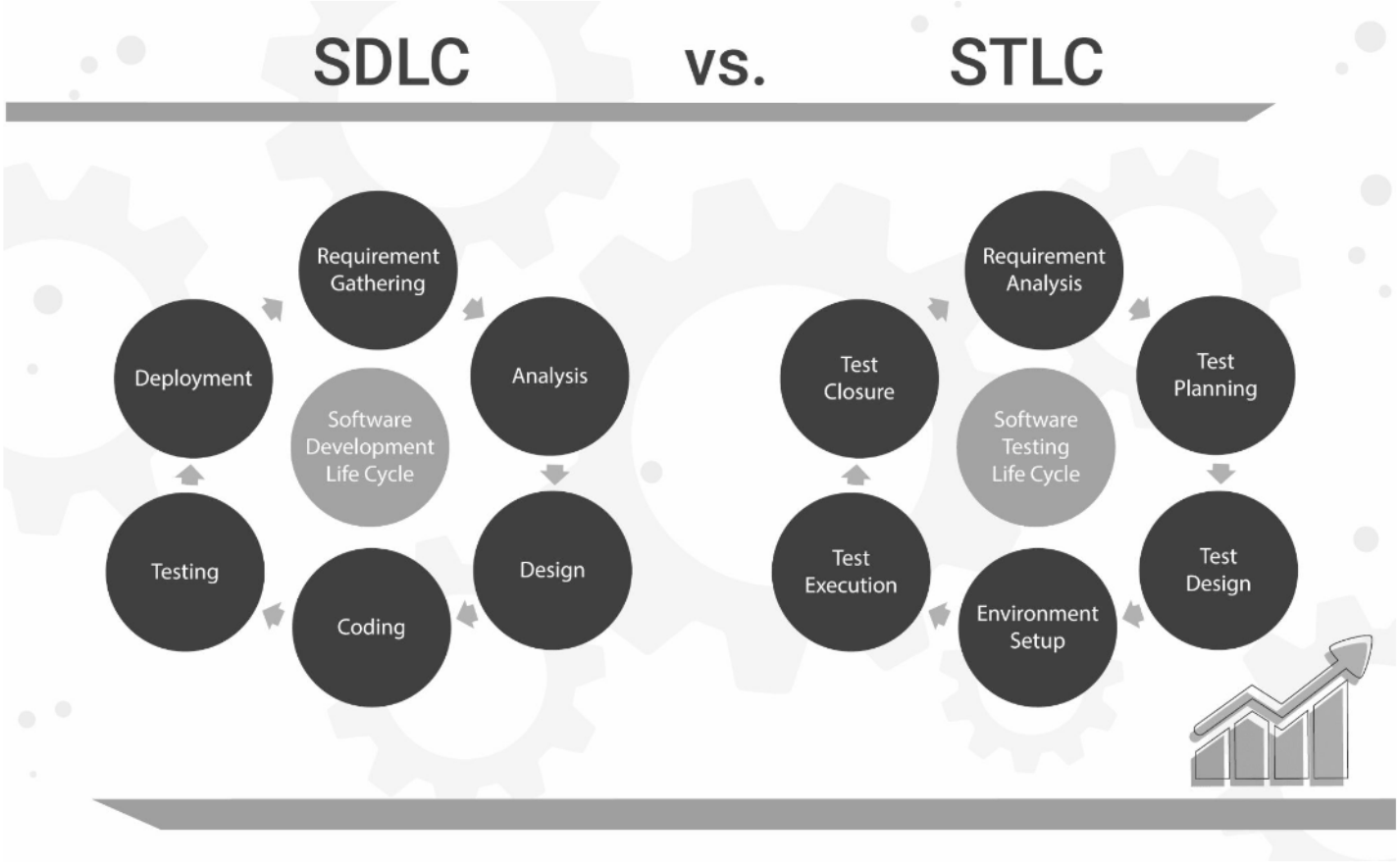
Test Execution: Documents - Bug Report , Test Execution Report

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.

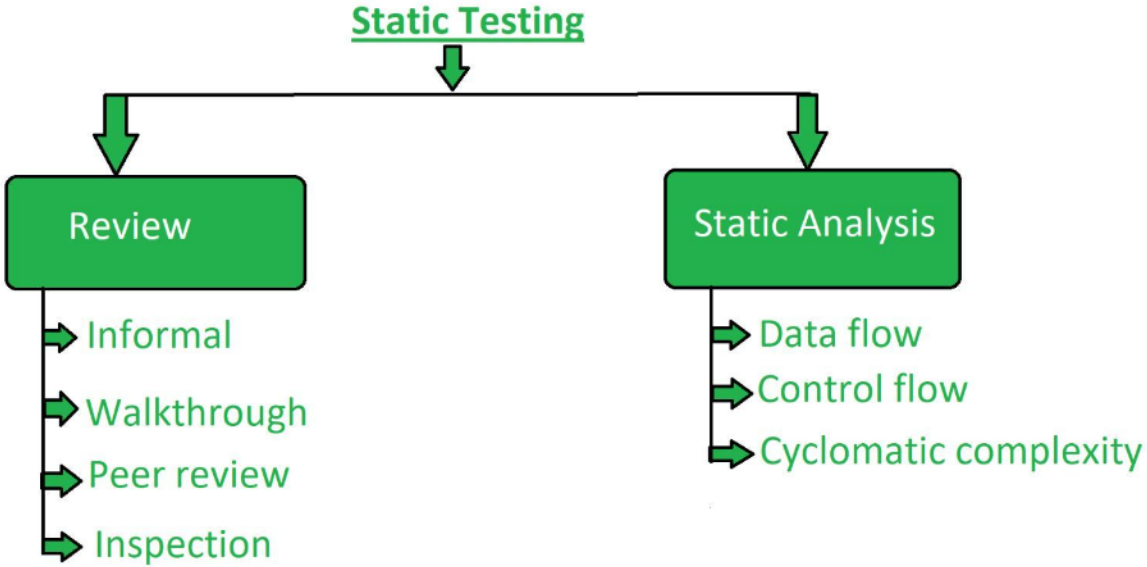

Setup Test Environment
Generate Test Cases
Generate Testing Data
Execute Test Cases
Defect Reporting
Regression Testing
Log Defects
Retest


SDLC
vs.
STLC
Requirement Gathering
Analysis
Design
Coding
Testing
Deployment
Software Development Life Cycle
Requirement Analysis
Test Planning
Test Design
Environment Setup
Test Execution
Test Closure
Software Testing Life Cycle

Black Box Testing	White Box Testing
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.

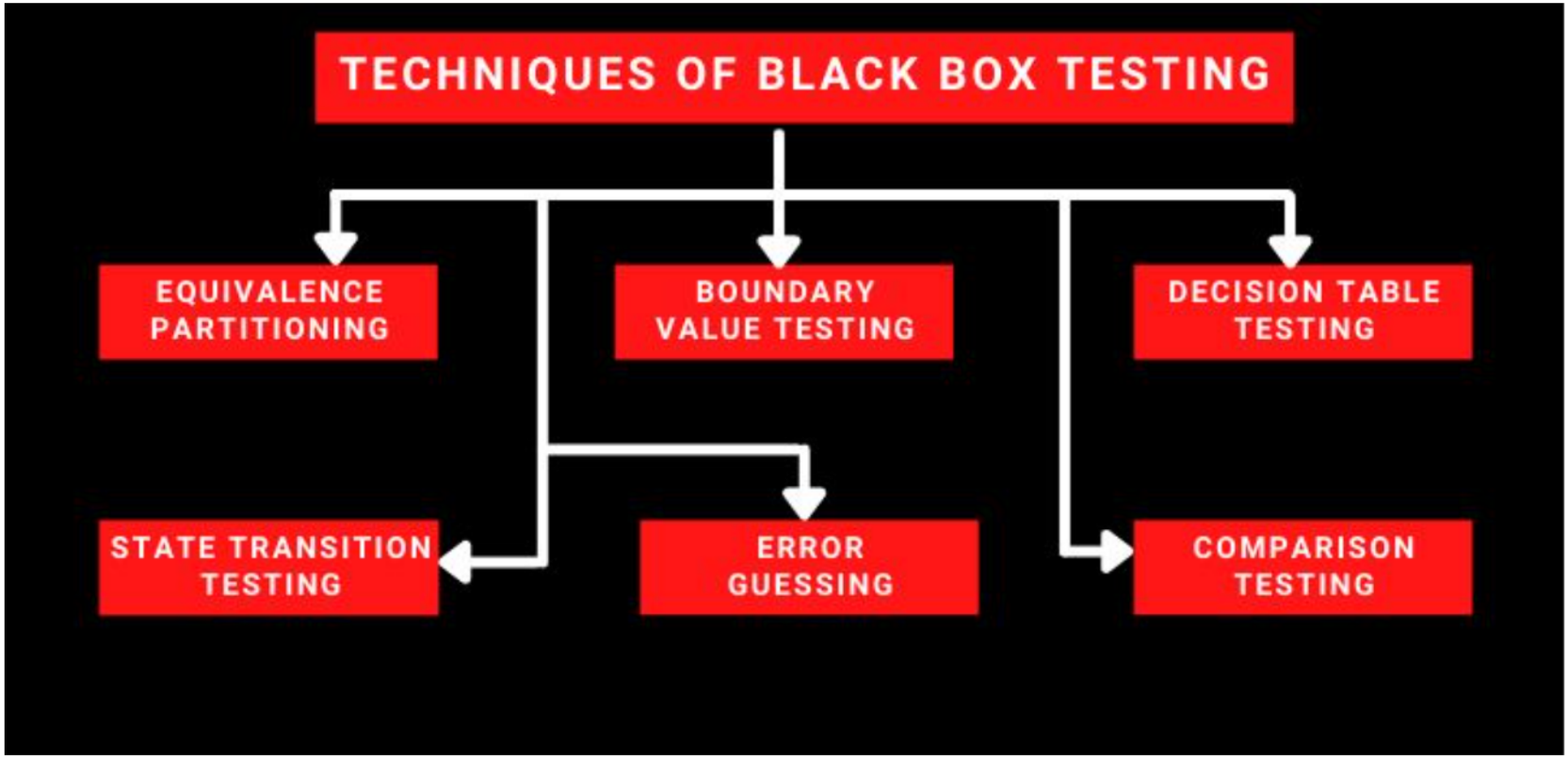


Static Testing

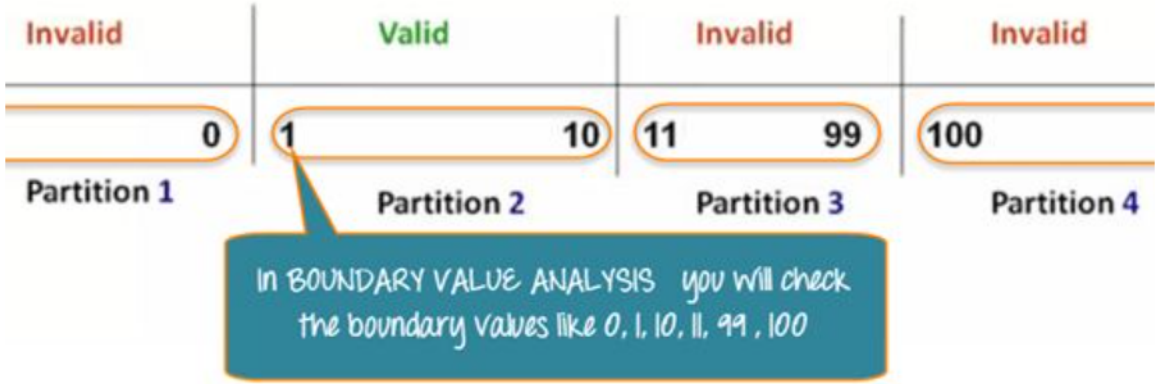
Static Analysis:
Static Analysis includes the evaluation of the code quality that is written by developers. Different tools are used to do the analysis of the code and comparison of the same with the standard.
It also helps in following identification of following defects:

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

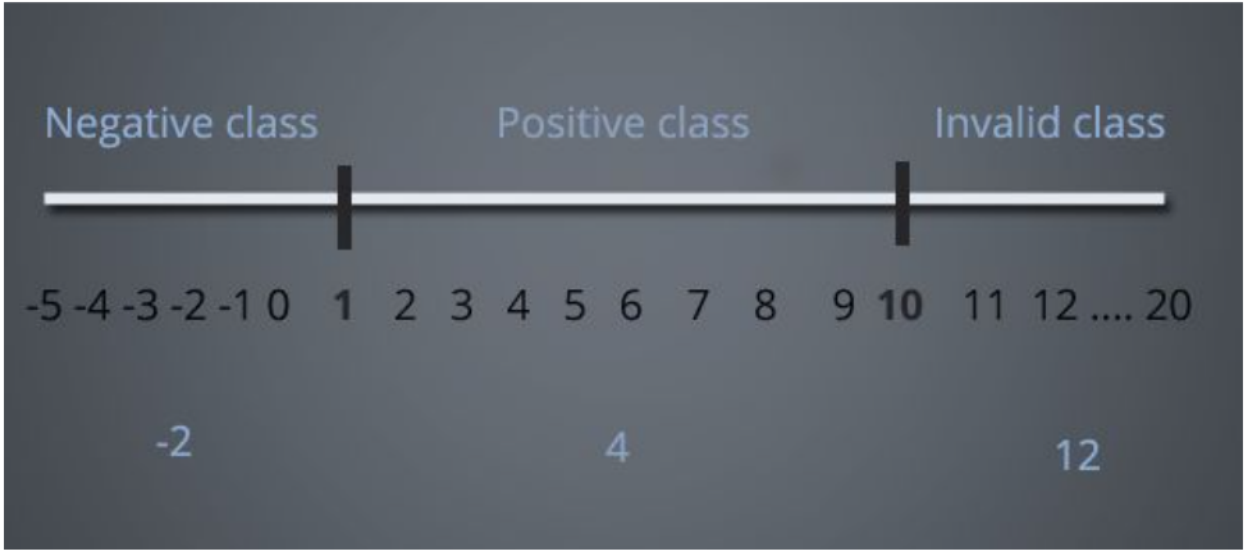
Static Testing	Dynamic Testing
It is performed in the early stage of the software development.	It is performed at the later stage of the software development.
In static testing whole code is not executed.	In dynamic testing whole code is executed.
Static testing prevents the defects.	Dynamic testing 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.


TECHNIQUES OF BLACK BOX TESTING
EQUIVALENCE PARTITIONING
BOUNDARY VALUE TESTING
DECISION TABLE TESTING
STATE TRANSITION TESTING
ERROR GUESSING
COMPARISON TESTING

Boundary Value Analysis.



Equivalence Class Partitioning



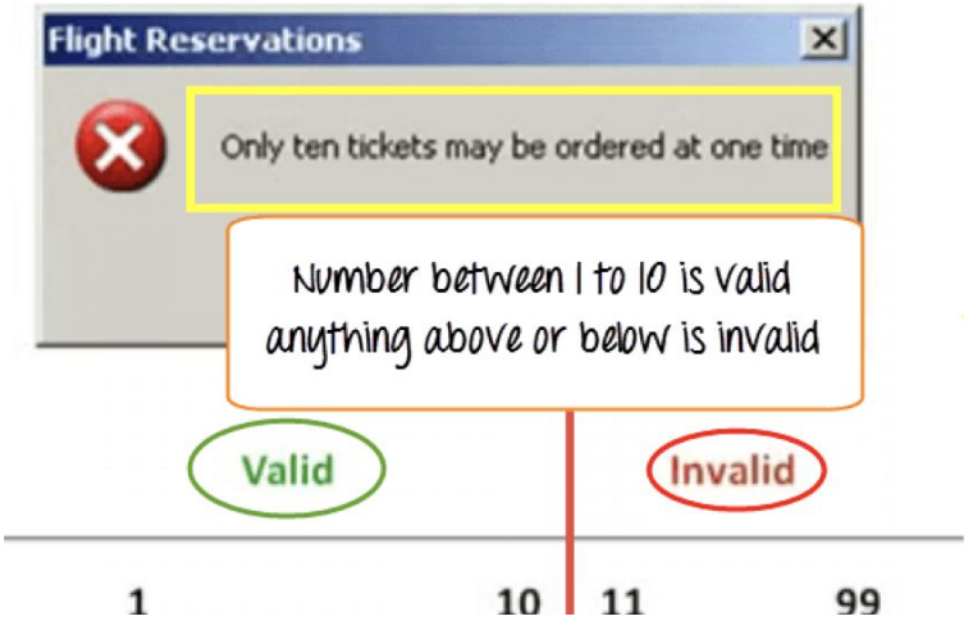
Practical Test cases

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

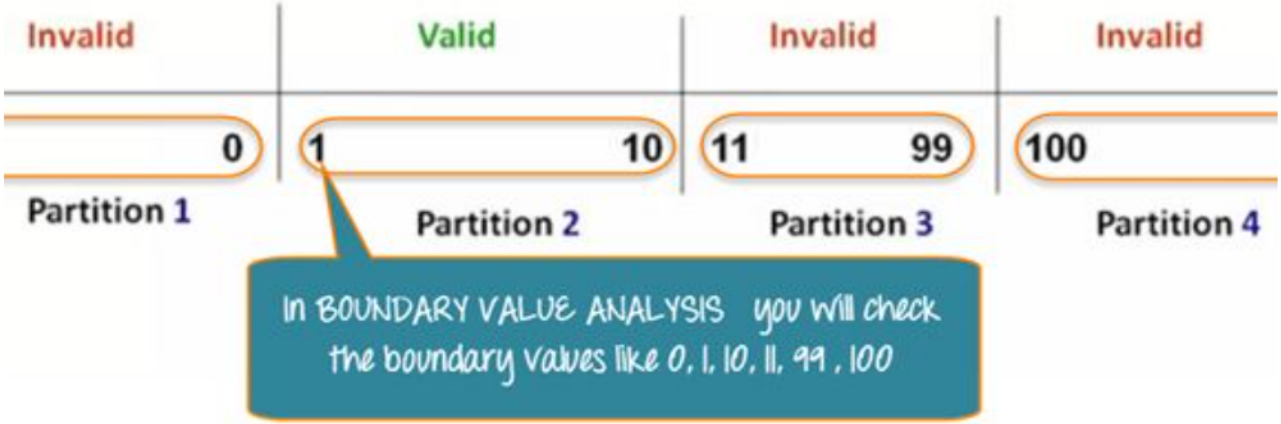


Equivalence partitioning and boundary value analysis(BVA) are closely related and can be used together at all levels of testing.

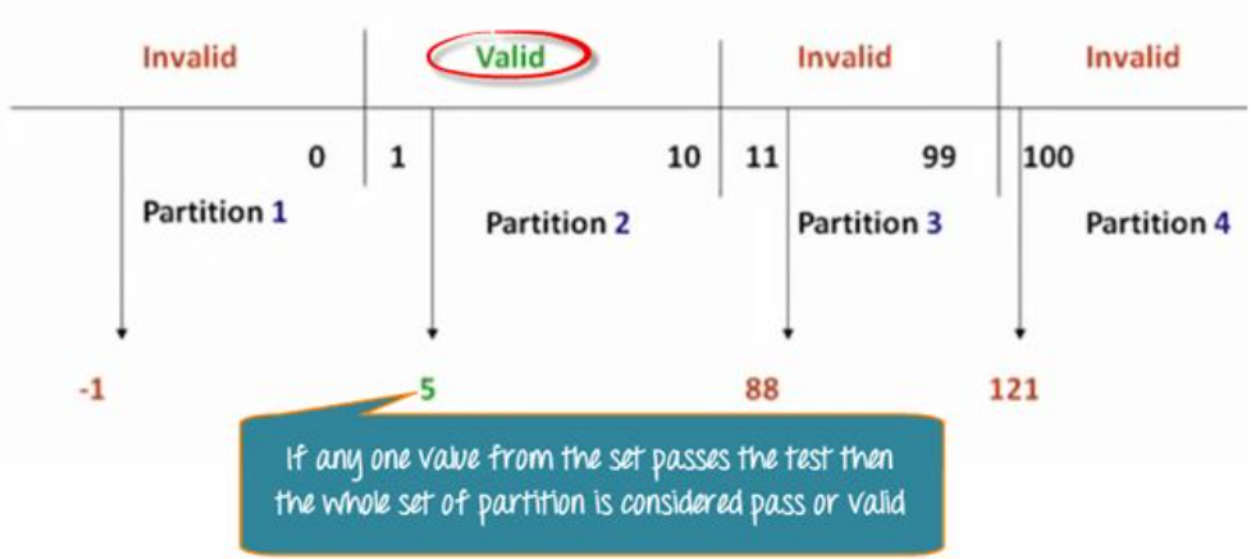
Problem

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**"

Problem


Invalid
Valid
Invalid
Invalid
0
1
10
11
99
100
Partition 1
Partition 2
Partition 3
Partition 4
In BOUNDARY VALUE ANALYSIS you will check the boundary values like 0, 1, 10, 11, 99 , 100

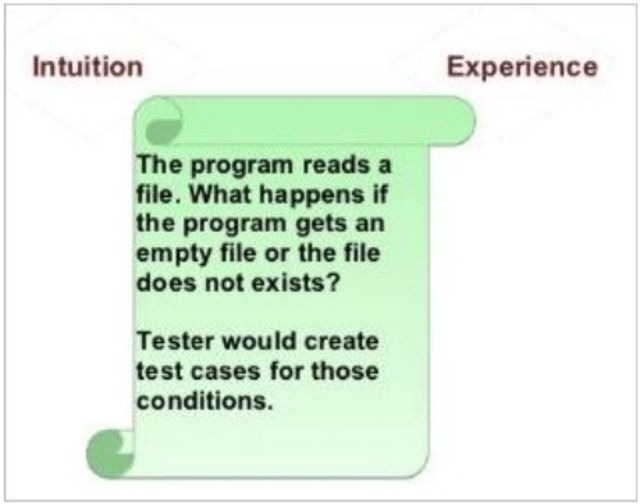
Problem


Invalid
Valid
Invalid
Invalid
0
1
10
11
99
100
Partition 1
Partition 2
Partition 3
Partition 4
-1
5
88
121
If any one value from the set passes the test then the whole set of partition is considered pass or valid

Decision Table based testing

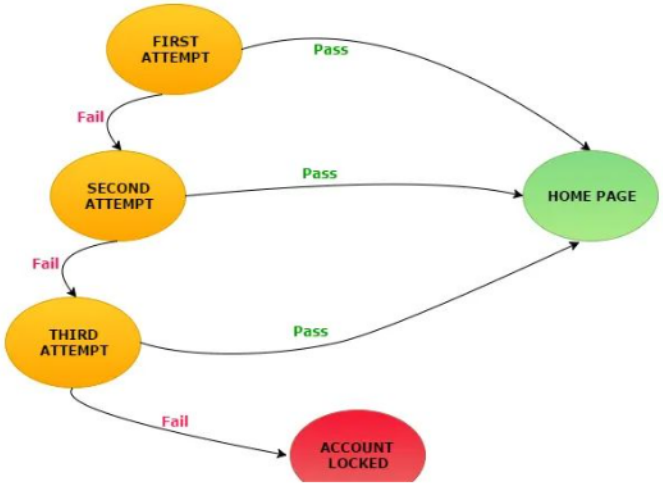
ID	CONDITIONS/ACTIONS	TEST CASE 1	TEST CASE 2	TEST CASE 3	TEST CASE 4
Condition 1	Valid User ID	T	T	F	F
Condition 2	Valid Password	T	F	T	F
Action 1	Home Page	Execute			
Action 2	Show a Message as 'Invalid User Credentials'		Execute	Execute	Execute

Error Guessing



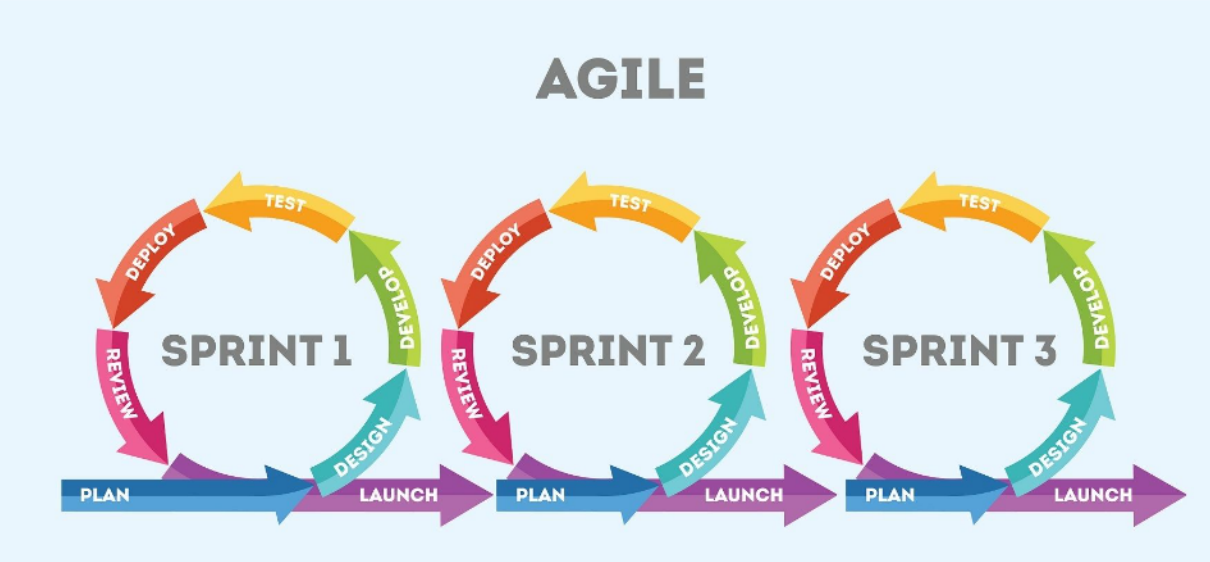
State Transition

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

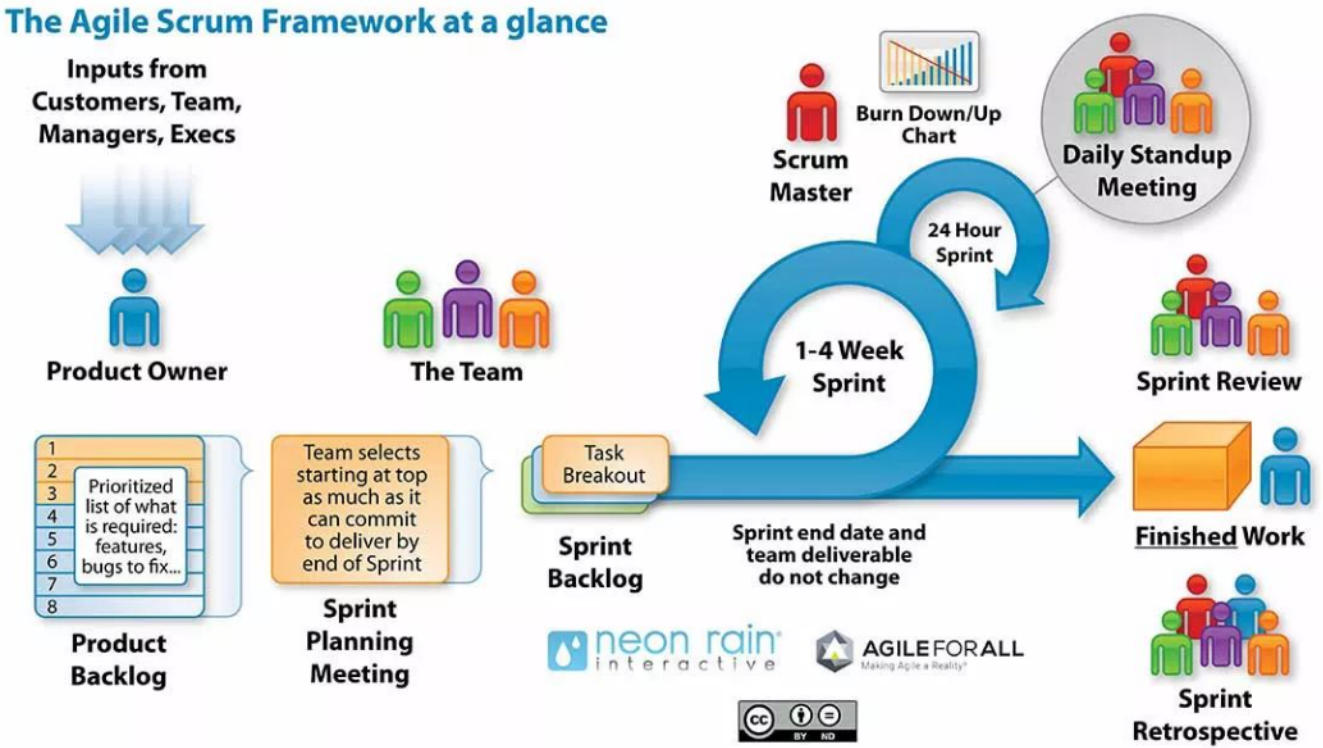
Good design

Measure progress

Continue seeking result

Reflect and adjust regularly

Agile


The Agile Scrum Framework at a glance
Inputs from Customers, Team, Managers, Execs
Product Owner
The Team
Scrum Master
Burn Down/Up Chart
Daily Standup Meeting
24 Hour Sprint
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Prioritized list of what is required: features, bugs to fix...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
neon rain interactive
AGILE FOR ALL
Sprint Retrospective