Software Quality Assurance

By SecureTest Partner

Books

Software Quality Assurance from Theory to Implementation

by Daniel Galin

Software Quality Engineering: Testing, Quality Assurance, & Quantifiable Improvement

by Jeff Tian

Ice Breaking 5 min every student

01 You career goal in SQA

02 SQA, Software Testing

03 Importance of SQA and Software Testing

04 Core objective of SQA

What is SQA

SQA aims to prevent defects, identify and correct them early in the development process, and establish a framework for continuous improvement. It is a proactive approach to quality management that contributes to the delivery of high-quality software products that meet user expectations and business requirements.

What is Software Testing?

Software testing is a crucial phase in the software development lifecycle that involves the evaluation of a software application to ensure that it behaves as expected and meets the specified requirements. The primary objective of software testing is to identify defects, errors, or bugs in the software and to ensure that the software functions correctly and efficiently.

Testing Principles

01 ØTesting Shows the presence of defects

02 ØEarly Testing

03 ØExhaustive Testing in Not Possible

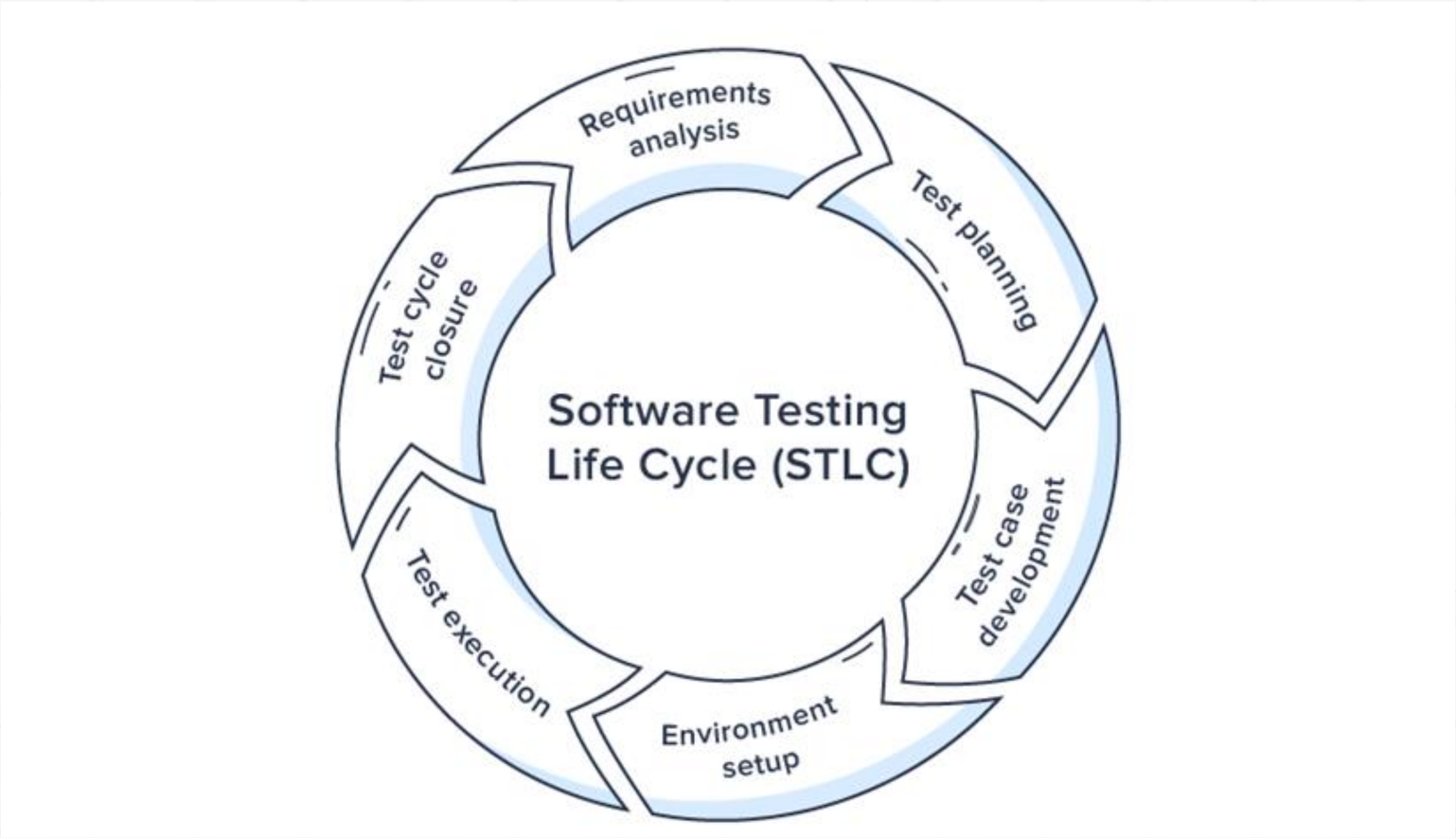
04 ØDefect Clustering

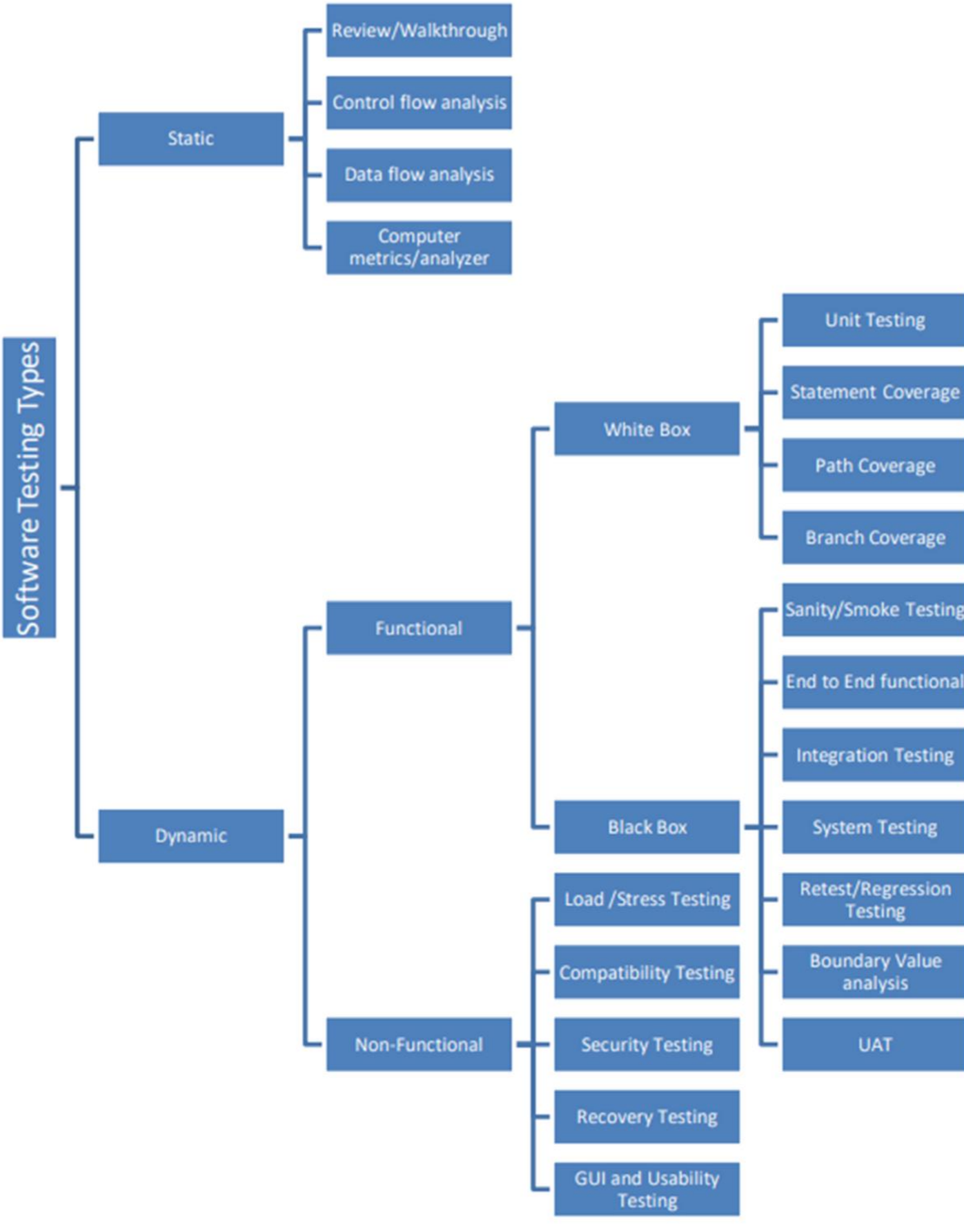
05 ØPesticide Paradox

06 ØTesting is context-dependent

7 ØAbsence of errors fallacy

Software Testing Life Cycle (STLC)




Software Testing Types
Static
Review/Walkthrough
Control flow analysis
Data flow analysis
Computer metrics/analyzer
Dynamic
Functional
White Box
Unit Testing
Statement Coverage
Path Coverage
Branch Coverage
Black Box
Sanity/Smoke Testing
End to End functional
Integration Testing
System Testing
Retest/Regression Testing
Boundary Value analysis
UAT
Non-Functional
Load /Stress Testing
Compatibility Testing
Security Testing
Recovery Testing
GUI and Usability Testing

5 important software testing techniques:

- Boundary Value Analysis (BVA)
- Equivalence Class Partitioning
- Decision Table based testing.
- State Transition
- Error Guessing

Boundary Value Analysis (BVA)

Boundary Value Analysis(Age accepts 18 to 56)		
Invalid (min-1)	Valid (min, min + 1, nominal, max – 1, max)	Invalid (max + 1)
17	18, 19, 37, 55, 56	57

Equivalence Class Partitioning

Search

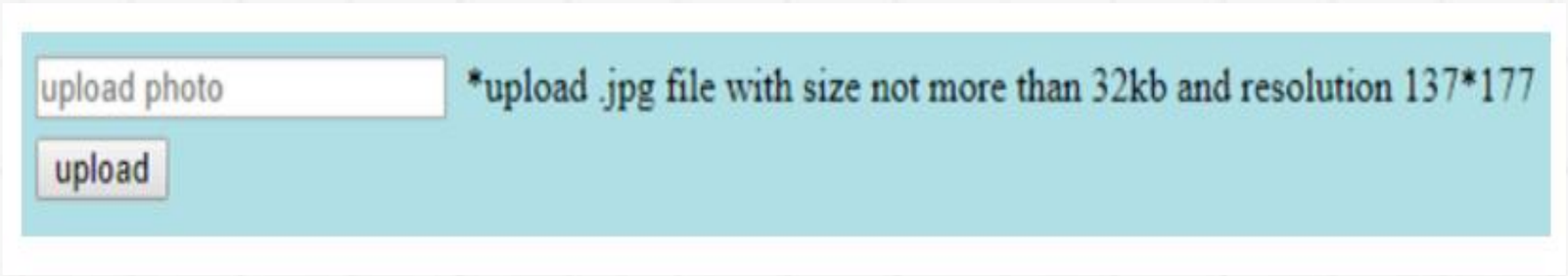
Equivalence Partioning		
Invalid	Invalid	Valid
77	84	45

Equivalence Class Partitioning

Enter OTP [ ] *Must include six digits

Equivalence Partioning			
Invalid	Invalid	Valid	Valid
Digits>=7	Digits<=5	Digits=6	Digits=6
67545678	9754	654757	213309

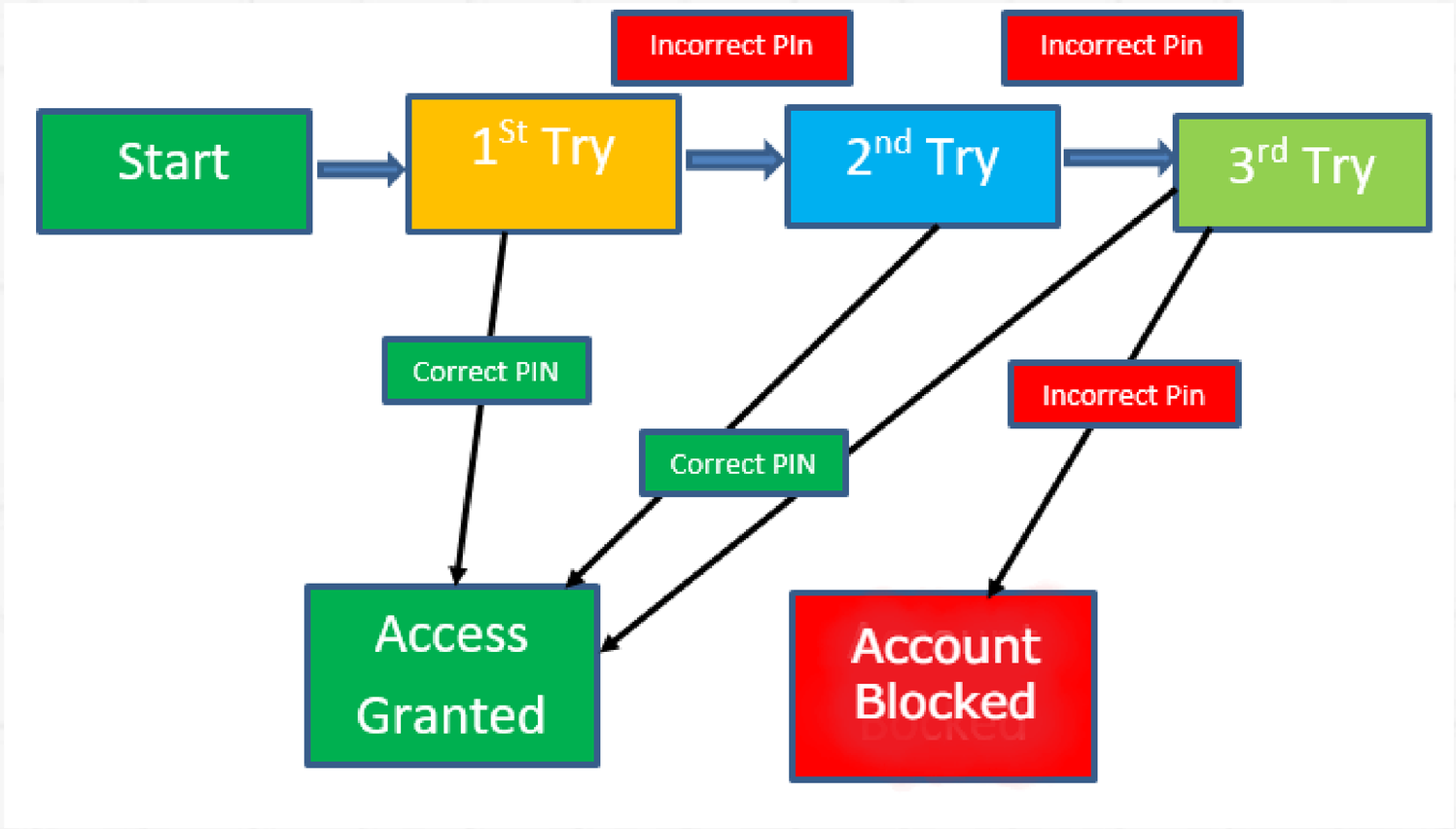
Decision Table Based Testing



Decision Table Based Testing

Conditions	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Format	.jpg	.jpg	.jpg	.jpg	Not .jpg	Not .jpg	Not .jpg	Not .jpg
Size	Less than 32kb	Less than 32kb	>= 32kb	>= 32kb	Less than 32kb	Less than 32kb	>= 32kb	>= 32kb
resolution	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177
Output	Photo uploaded	Error message resolution mismatch	Error message size mismatch	Error message size and resolution mismatch	Error message for format mismatch	Error message format and resolution mismatch	Error message for format and size mismatch	Error message for format, size, and resolution mismatch

State Transition



Error Guessing

Factors used in error guessing :

- Lessons learned from past releases.
- Experience of testers.
- Historical learning.
- Test execution report.
- Earlier defects.
- Production tickets.
- Normal testing rules.
- Application UI.
- Previous test results.

Common Testing Strategies

01 Manual Testing

02 Automation Testing

03 Risk-Based Testing

04 Smoke Testing

05 Sanity Testing

06 Regression Testing

07 Acceptance Testing

08 Usability Testing

09 Exploratory Testing

10 Incremental Testing

11 Ad Hoc Testing

12 A/B Testing

Functional Testing

Functional testing is a type of software testing that focuses on verifying that the software application or system functions as intended and meets the specified requirements. It involves evaluating the functional aspects of the software by testing its individual components, features, and the overall system to ensure that it behaves correctly under various conditions. The goal of functional testing is to ensure that the software application delivers the expected functionality to end-users.

Non-Functional Testing

Non-functional testing is a type of software testing that focuses on the non-functional aspects of a software application rather than its specific functionalities. It is designed to assess how well the software performs under certain conditions and to measure its qualities related to aspects such as performance, reliability, scalability, usability, and security. Unlike functional testing, which verifies specific features and functions, non-functional testing evaluates the characteristics that impact the overall user experience and the software's ability to meet certain criteria.

Manual Testing

- Human testers manually execute test cases without the use of automation toolss.
- Suitable for exploratory testing, intuitive testing based on human judgment.
- Time-consuming for repetitive testing, prone to human errors.

Automation Testing

Testing involves automated tools and scripts to execute test cases.

- Advantages: Efficient for repetitive and large-scale testing, provides quick feedback.
- Challenges: Initial setup time, not suitable for exploratory testing, maintenance can be resource-intensive.

Unit Testing

Testing individual units or components of a software application in isolation.

- Purpose: Verify that each unit functions correctly as designed.
- Tools: JUnit (Java), NUnit (.NET), pytest (Python).

Acceptance Testing

- Testing to ensure that the software meets user acceptance criteria.
- Types: User Acceptance Testing (UAT), Alpha Testing, Beta Testing.

Regression Testing

- Verifying that new changes do not negatively impact existing functionalities.
- Purpose: Ensure that existing features continue to work as intended.

Smoke Testing

- Initial testing to ensure that critical functionalities of a software build work as expected.
- Purpose: Quickly identify major issues early in the development process.

Exploratory Testing

- Simultaneous learning, test design, and execution.
- Purpose: Discovering defects that are not covered by existing test cases.

Usability Testing

- Evaluating the user-friendliness and user experience of the software.
- Purpose: Ensure the software is easy to use and meets user expectations.

Risk-Based Testing

- Description: Allocating testing efforts based on perceived risks.
- Purpose: Optimize testing resources and focus on high-risk areas.

White Box Testing

- Testing that involves understanding and testing the internal logic and structure of the software's code.
- Purpose: Verify the correctness of the internal code structure and logic.

Black Box Testing

- Testing without knowledge of the internal code or logic, focusing on testing the software's external functionality.
- Purpose: Validate that the software meets specified requirements without knowledge of the internal implementation.

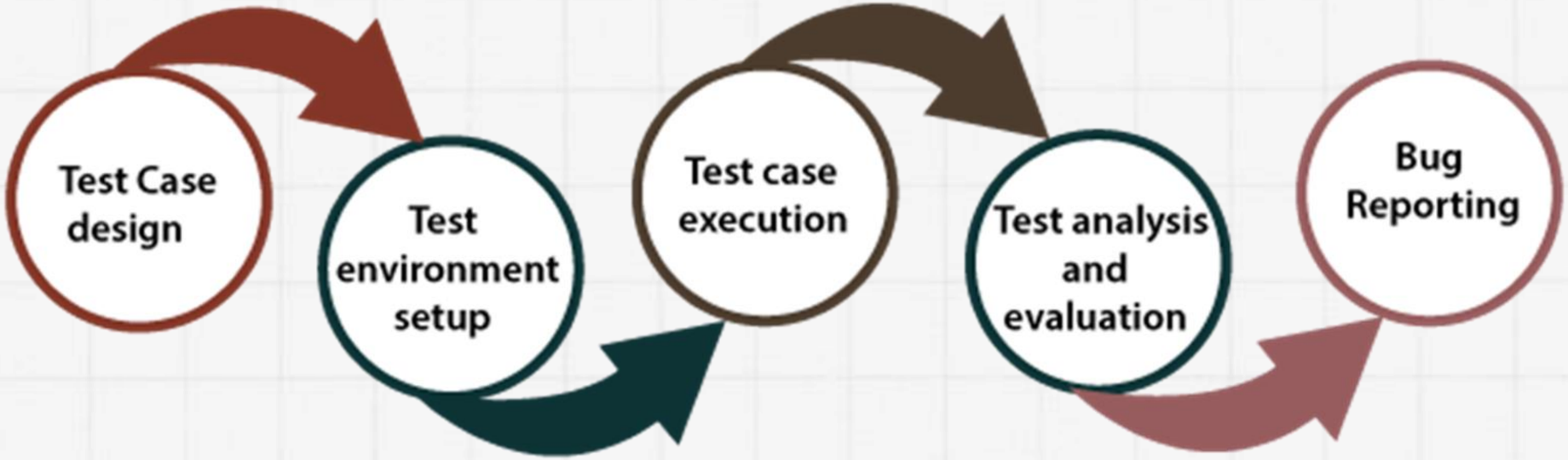
A/B Testing

- Comparing two versions of a software application by deploying different versions to different user groups and analyzing the performance.
- Purpose: Determine which version performs better in terms of user engagement or other specified metrics.

Sanity Testing

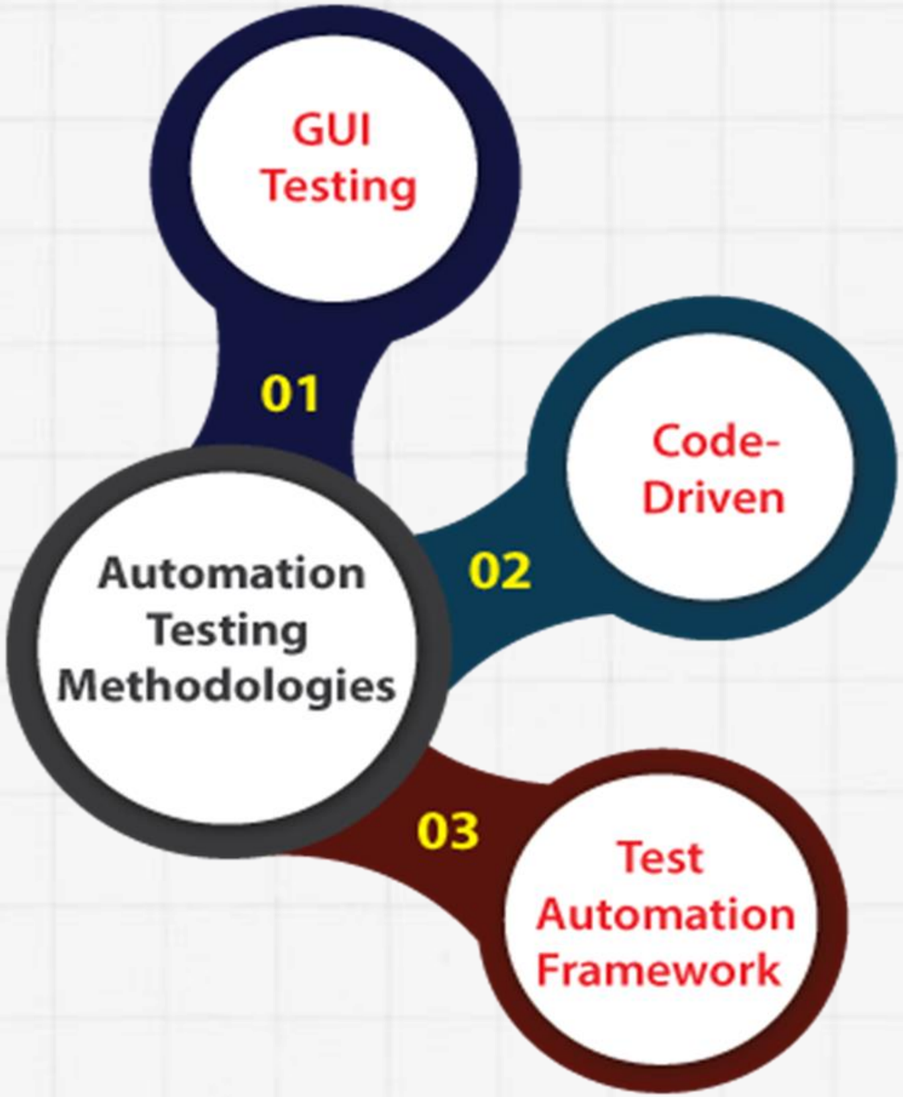
Sanity testing, also known as sanity check or build verification testing, is a type of software testing that quickly evaluates whether a specific set of functionalities or components of a software application are working as intended after a change, addition, or bug fix. The purpose of sanity testing is to ensure that the recent modifications have not adversely affected the existing functionalities and that the software is stable enough for further, more detailed testing.

TESTING PROCESS



Why Automation

Automation Testing Methodology



Tools

01 Test management tool

02 Bug tracking tool

03 Automated testing tool

04 Performance testing tool

05 Cross-browser testing tool

06 Integration testing tool

07 Unit testing tool

08 Mobile/android testing tool

09 GUI testing tool

10 Security testing tool

Selenium

- This is an open source testing tool.
- It can be used for both website gui test and cross browser testing.


Postman


Postman is a popular collaboration platform for API development, testing, and documentation. It provides tools for building, testing, and managing APIs, making it easier for developers to work with APIs in their projects. Here are some key features and functionalities of Postman:

Jira

Jira is a popular project management and issue tracking tool developed by Atlassian. It is widely used by software development teams to plan, track, and manage their projects. Jira provides a flexible and customizable platform that can be adapted to various project management methodologies, including Agile, Scrum, and Kanban.

Jmeter

Apache JMeter is an open-source software testing tool designed for performance testing and functional testing of web applications. It provides a graphical user interface (GUI) and a set of features that allow users to simulate various scenarios and measure the performance of web applications under different conditions.

Thank you!

By SecureTest Partners