

SOFTWARE TESTING CONCEPTS



SOFTWARE TESTING ???

"Software testing is a way to assess the quality of the software and to reduce the risk of software failure in operation."

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.



WHY WE NEED TO TEST?

For improved Product Quality

Its very Cost Effective

Non-Functional requirements like Security, Performance

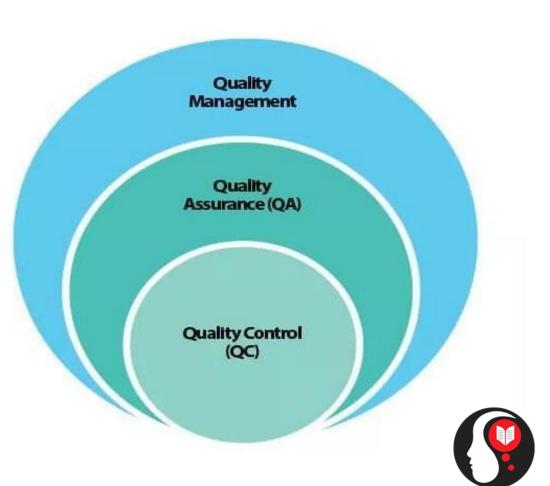
For better Customer Satisfaction



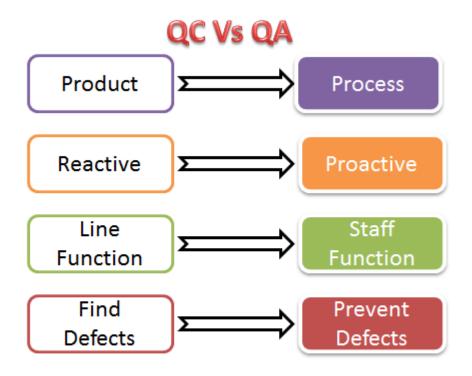
QUALITY MANAGEMENT

Quality Assurance is **static testing**. In other words, it comes under "Verification" with the primary purpose of the prevention of defects

Quality Control is **dynamic testing**. In other words, it comes under "Validation" with the primary purpose of the identification of defects



QUALITY ASSURANCEVS QUALITY CONTROL



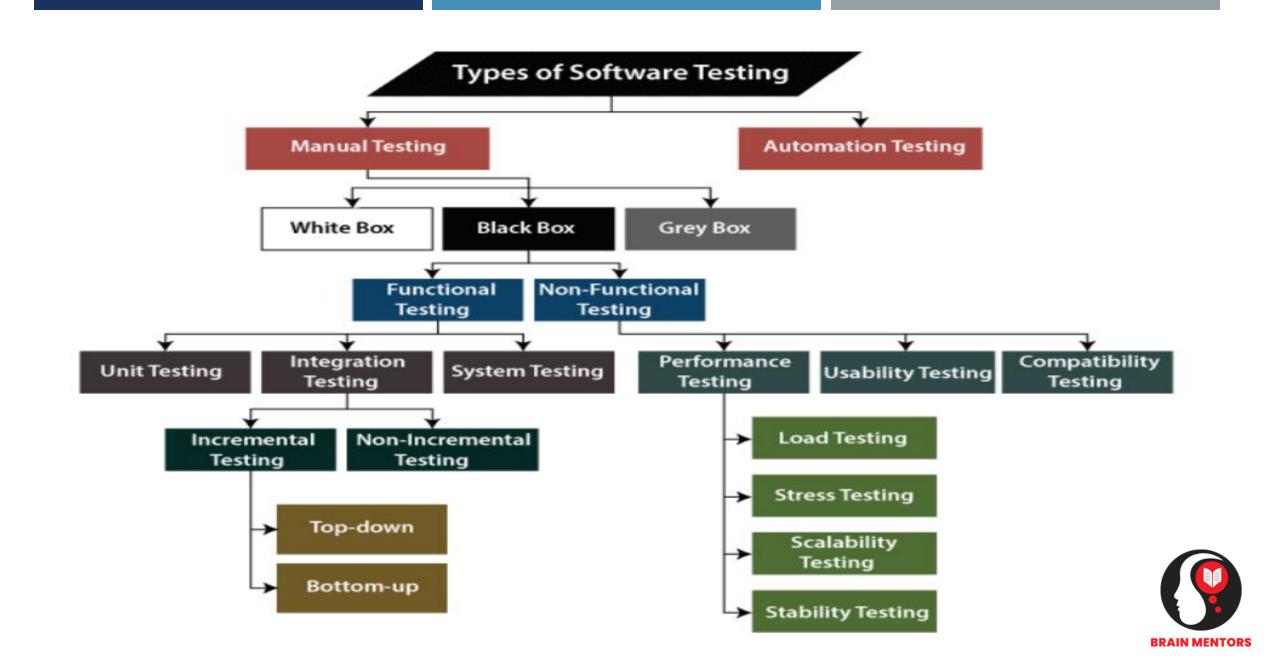
Quality Control Vs Quality Assurance



COMMON INTERVIEW QUESTIONS

- Quality Assurance vs Quality Control
- Difference Between Test Scenarios, Test Cases and Test Scripts.
- What are different Levels of Testing.
- What are different Types of Testing.
- Define Test Plan.
- Define UAT and its Types.
- Write Test Cases for an Application
- Defect Life Cycle
- Priority and Severity
- Automation Know how





FUNCTIONAL AND NON-FUNCTIONAL TESTING

Posting a message on Facebook page

FUNCTIONAL

WHAT a system does

NONFUNCTIONAL

HOW well
the system does it
within design &
resource
constraints

No data leakage, Millions of users can access in parallel

Keeping the Actual User in Mind

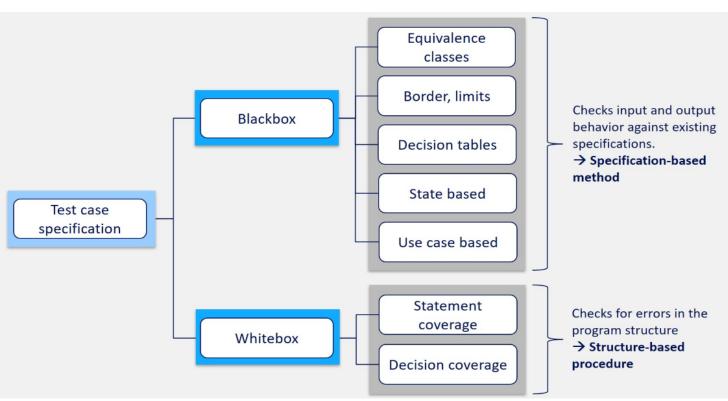


TESTING TECHNIQUES

Equivalence Class Partitioning

Boundary Values

Use Case Based





EXAMPLES OF TECHNIQUES

Use Case Based

- Login
- Change Password
- Set FB Status
- ATM Transaction

Equivalence Class Partitioning

- Enter Age
- Credit Card Expiry Date
- ATM Transaction

Boundary Values

- Valid and Invalid Boundaries
- Invalid Boundary Check { Min-I; Max+I}
- Valid BoundaryCheck {Min; Min+I; Max-I; Max }



SOFTWARE TESTING TYPES



BASIC OF SOFTWARE TESTING



- 1. BLACK BOX
- 2. WHITE BOX
- 3. GREY BOX



LEVEL OF SOFTWARE TESTING



- 1. UNIT TESTING
- 2. INTEGRATION TESTING
- 3. SYSTEM TESTING
- 4. USER ACCEPTANCE TESTING



CATEGORIES OF SOFTWARE TESTING



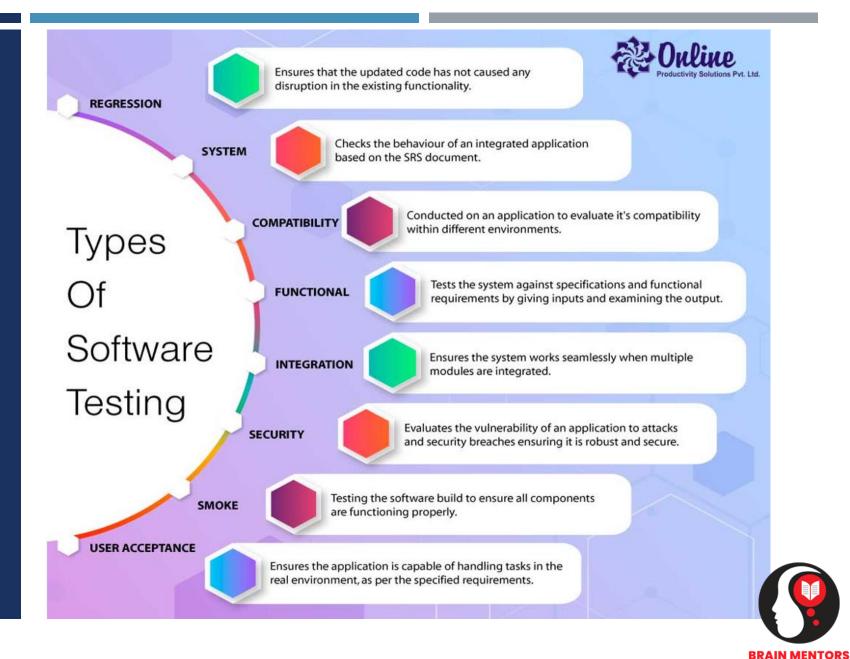
- 1. GUI
- 2. FUNCTIONAL
- 3. VALIDATION
- 4. SECURITY
- 5. SMOKE AND SANITY
- 6. REGRESSION
- 7. MONKEY
- 8. GORILLA
- 9. AGILE
- 10. USABILITY
- 11. EXPLORATORY
- 12. ADHOC
- 13. STATIC AND DYNAMIC
- 14. COMPATIBILITY
- 15. STRESS/LOAD TESTING
- 16. PERFORMANCE
- 17. VOLUME TESTING

TESTING CATEGORIES TO REMEMBER

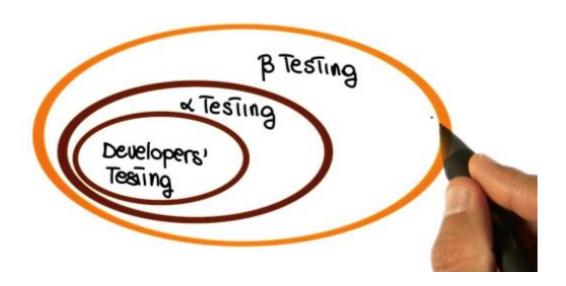


SOME COMMON TESTING TYPES...





UAT TESTING TYPES



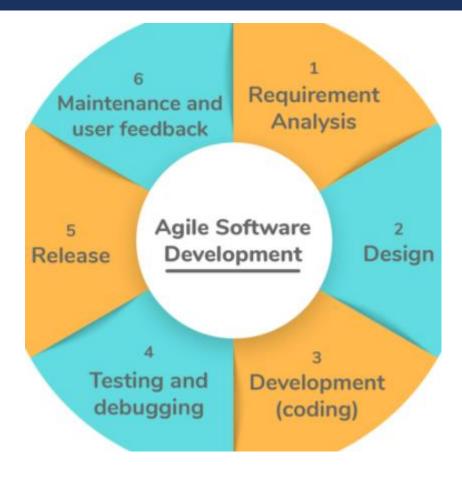
Beta Alpha Test every user journey and Understand how real users interact Main purpose ensure the application is working with the application and test how it as expected. works in real-world conditions. Internal team, potentially also Joe Public. Either via invite or via Done by end-customer. controlled release. White box. The testers know Black box. The testers just see Visibility what is happening and what they the application and any are testing. associated help or tool-tips. Rigorously structured. Every flow Completely unstructured. Users are is tested and all results recorded free to do what they want. Feedback Structure and carefully analyzed. is requested, but not required. Not interested in how the Reliability and performance are key Performance application or backend performs, aspects of beta testing. Along with simply in functionality. application security and stability. Any bugs identified are fixed. Any Show-stopper bugs are fixed. A/B suggestions for small changes are testing may be conducted and result in Post-test-actions analyzed and may be implemented. a decision. But, in general, the results of E.g. minor UI changes. feedback are used to influence the next version. This can be a long process, Usually, this is a short process however, you should aim to keep it lasting a few weeks. However, Duration as short as possible (to avoid some companies run betas that

delays with your release).

last for many years.



AGILE DEVELOPMENT LIFECYCLE





SOFTWARE TESTING LIFECYCLE

Requirement Analysis

Test Planning

Test Case
Designing
and
Development

Test Environment Setup

Test Execution

Test Closure



TEST PLANNING





TEST SCENARIOS/CASES AND SCRIPTS

Test Scenarios: A Test Scenario is any functionality that can be tested. It is also called Test Condition or Test Possibility.

Test Cases: It is a document that contains the steps that have to be executed; it has been planned earlier.

Test Script: It is written in a programming language and it's a short program used to test part of the functionality of the software system. In other words, a written set of steps that should be performed manually.



Test case ID	Test case description	Prerequisites	Test steps	Test data	Expected Result	Actual Result	Status	Created By	Date of creation	Executed By	Date of execution
TC001	The objective of this	1. User is authorized	1. Enter valid username	1. User	1. User should	1. If the valid	Fail	Rajesh	1/1/2016	Umesh	1/2/2016
	test case is to verify	2. Has an account in	2. Enter valid password	account	be able to login	credentials	369	1,050,000	ANA GOOGSCALLY		1020005865
	the 'Login' of Gmail	Gmail	3. Click on 'Login' button	should	his Gmail	are entered					
	account	0.500.00		be	account with his	then the user					
				present	valid credentials	will be able					
				in Gmail	CONTRACTOR OF THE PARTY OF THE	to login his /					
				1 22 - 25	2. 'Invalid	her account					
					username or						
					password'	2. If invalid					
					should get	credentials					
					displayed if the	are entered					
					username and	then nothing					
					password are	happens(the					
					not valid	expected					
					DOS TRONIBOS	message is					
						not					
						displayed)					



CATEGORIES TO KEEP IN MIND WHILE WRITING TEST SCENARIOS/CASES

Performance Testing
Stress Testing
Failover Testing
Compatibility Testing
Usability Testing
Scalability Testing
Security Testing



Reliability Testing
Endurance Testing
Recovery Testing
Internationalization
Testing
Localization Testing
Accessibility Testing
Compliance Testing

BRAIN MENTORS

EXAMPLES OF TESTING TYPES

Social Media Applications

- Performance Testing
- Stress Testing
- Security Testing
- Internationalization Testing
- Localization Testing
- Accessibility Testing

Helpline System

- Performance Testing
- Stress Testing
- Reliability Testing

Books Management System

Performance Testing



• Mobile Recharge via Paytm vs Banking Applications



QA BEST PRACTICES

Test one thing at a time

Understand the types of testing on offer

Use existing set of Regression Test

Report and track bugs

Choose the right environment for tests

Don't neglect the UI

Automate, Automate & Automate where due



Severity

Severity determines
The defect's effect on the application.

How bad the defect is

Severity is given by QA testers

Levels

- · Critical: the software will not run
- High: unexpected fatal errors (includes crashes and data corruption)
- · Medium: a feature is malfunctioning
- · Low: a cosmetic issue.

Priority

Priority determines
The defect urgency of repair.

How soon we need to fix

Priority is given by Test lead or project manager.

Levels

- . P1: fix before next build to test
- · P2: fix before final release
- P3: we probably won't get to these, but we want to track them anyway to resolve the priority-severity divide.

PRIORITY AND SEVERITY



PRIORITY

SEVERITY

HIGH

LOW

E E

Key features failed and no workaround **E.g.** Login button is not working Basic feature failed but it has a huge impact on customer's business **E.g.** Misspelled Company logo

NO7

Key features failed but there is no impact on customer's business **E.g.** Calculation fault in yearly report which end user won't use regularly

Cosmetic issues **E.g.** Font family mismatch in a report

PRIORITY AND SEVERITY



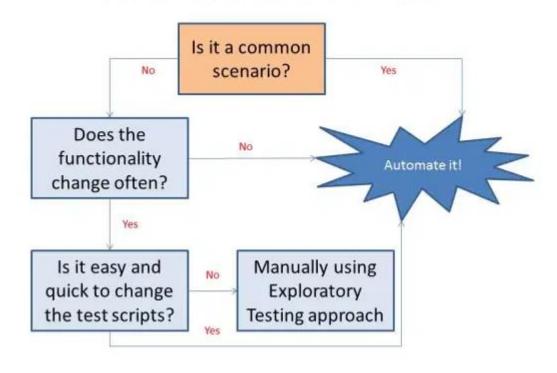
DEFECT LIFE CYCLE

New Assigned Open Fixed Ready QA in Progress Closed



WHAT TO AUTOMATE AND WHAT NOT

Shall I automate this test?





WHAT NOT TO AUTOMATE

It is best NOT to automate:

- Subjective tests
- Frequently changing requirements
- · Ad-hoc tests



WHEN TO AUTOMATE

Smoke Testing

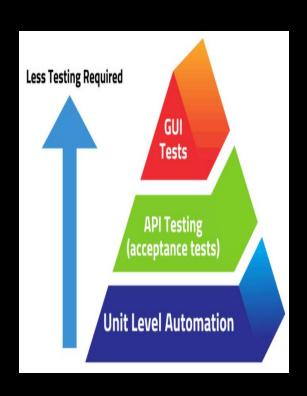
Regression Testing

Load Testing

Performance Testing



AGILE AUTOMATION PYRAMID



Layers of Agile Test Automation Pyramid



BRAIN MENTORS

Best Automation Testing Tools for 2021





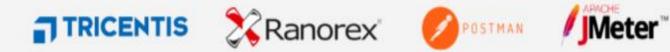


















HOW TO AUTOMATE WEB APPLICATIONS

- Selenium
- Python

Selenium automates browsers. That's it! What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.

Boring web-based administration tasks can (and should) also be automated as well.

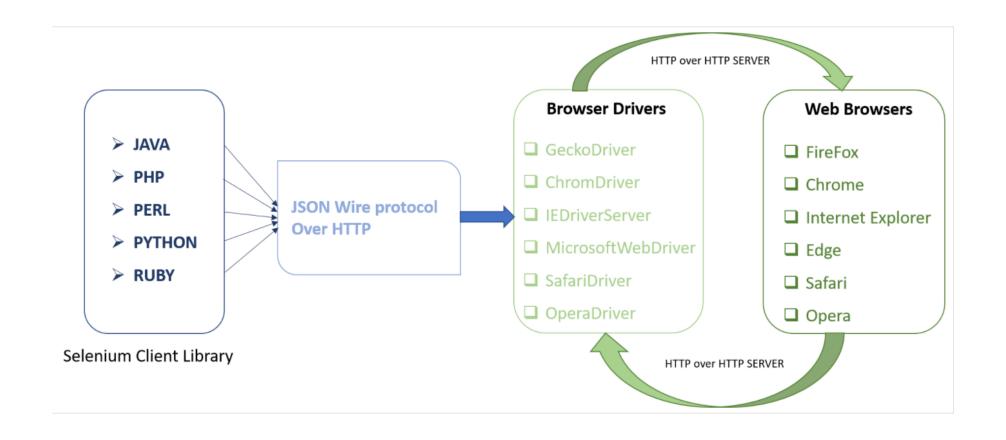








SELENIUM ARCHITECTURE





SELENIUM LOCATORS (8 IN TOTAL)

THE RECOMMENDATION IS TO KEEP YOUR LOCATORS AS COMPACT AND READABLE AS POSSIBLE.
ASKING WEBDRIVER TO TRAVERSE THE DOM STRUCTURE IS AN EXPENSIVE OPERATION, AND THE MORE YOU CAN NARROW THE SCOPE OF YOUR SEARCH, THE BETTER.

Locator	Description
class name	Locates elements whose class name contains the search value (compound class names are not permitted)
css selector	Locates elements matching a CSS selector
id	Locates elements whose ID attribute matches the search value
name	Locates elements whose NAME attribute matches the search value
link text	Locates anchor elements whose visible text matches the search value
partial link text	Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected.
tag name	Locates elements whose tag name matches the search value
xpath	Locates elements matching an XPath expression

BRAIN MENTORS