IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

Study and Analysis of Automation Testing

Ashish Mishra
Student, Department of Information Technology
Dronacharya College of Engineering, Gurugram, Haryana, India

Abstract: *Automation Testing means using an automation tool to execute your test case suite. The automation software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources. Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether. Testing is a very important activity in Software Development Process. It is to examine & modify source code. Effective Testing produces high quality software. This Paper deals with a significant and vital issue of Software Testing. Testing can be conducted manually as well as Automated. These Techniques have their own advantages & disadvantages. The Objective of this paper is to Introduce in Software Testing and Software Test Automation where we cover what is Test Automation, Why Automation Testing, which test cases to automate, Who should be involved in test automation, Common Misconceptions About Automated Testing, How Automation script works, Automation Testing Process, Why Framework for Automation, Types of Automation Framework, Automation at Different levels, Automation Testing Tools.*

Keywords: Software Testing, Agile in Software Testing, Test Case Design, Automation in DevOps.

I. INTRODUCTION

1.1 What is Test Automation?

Automation Testing means using an automation tool to execute your test case suite.
The automation software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources.
Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

1.2 Why Automation Testing?

Automated software testing is important due to the following reasons:

- Manual Testing of all workflows, all fields, all negative scenarios is time and money consuming
- It is difficult to test for multilingual sites manually
- Automation does not require Human intervention. You can run automated test unattended (overnight)
- Automation increases the speed of test execution
- Automation helps increase Test Coverage
- Manual Testing can become boring and hence error prone.

1.3 Which Test Cases to Automate?

Test cases to be automated can be selected using the following criterion to increase the automation ROI

- High Risk - Business Critical test cases
- Test cases that are repeatedly executed
- Test Cases that are very tedious or difficult to perform manually
- Test Cases which are time-consuming

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

The following category of test cases are not suitable for automation:

- Test Cases that are newly designed and not executed manually at least once
- Test Cases for which the requirements are frequently changing

1.4 Who should be involved in test automation?

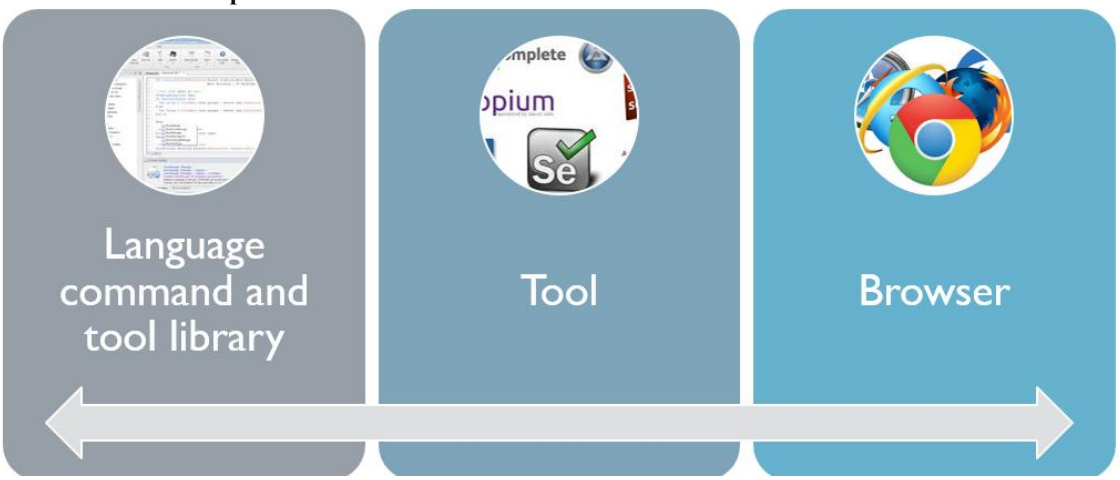
When evaluating a testing solution, it's important to have a tool that fits the needs of all of the different team members who will be involved in the testing process. These include:

- Manual testers: Record and replay is crucial for manual testers, especially those who are new to automation. Being able to use the same recorded script with variety of input data can come in handy while identifying and fixing problems across multiple environments.
- Automation engineers: For automation engineers, robust support for scripting languages, integrations with CI systems, and the ability to scale tests easily could be important.
- Developers: Implement testing into the development process requires the ability to conduct tests within IDEs such as Eclipse and Visual Studio**.**

1.5 Common Misconceptions about Automated Testing

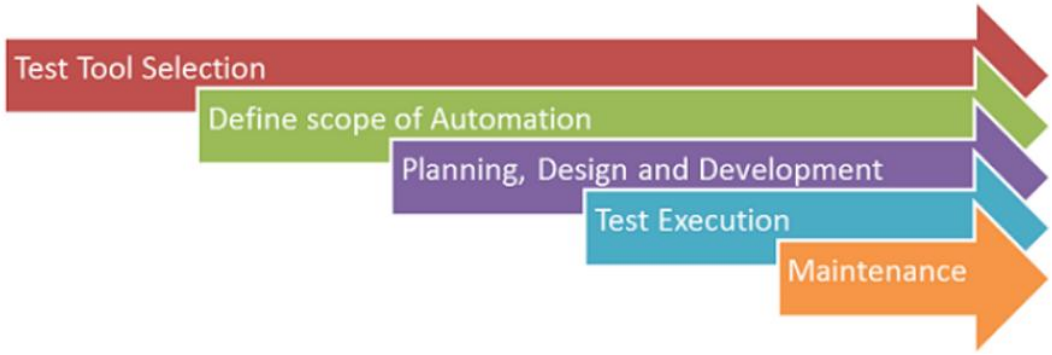
- **Automation will provide you with more free time:** The misconception that automated testing will give you more free time is both true and false. In manual testing, most of the time is devoted to exploratory and functional testing where you would manually search for errors. Once that process is complete, the manual tester must repeatedly go through the same steps over again.
- **The Cost of Automated Testing is Too High:** At first, the investment in automated testing might seem cost prohibitive, especially if you're a smaller company. But analysis has shown that, over time, automated testing pays for itself.
- **Automated Testing is Better Than Manual Testing:** The reality is, there is no "better" or "worse" in the automated vs. manual debate, there's just "different." Each approach has its own advantages and disadvantages. Manual testing is performed by a human sitting in front of a computer carefully going through application via SQL and log analysis, trying various usage and input combinations, comparing the results to the expected behavior and recording the results. Automated testing is often used after the initial software has been developed. Lengthy tests that are often avoided during manual testing can be run unattended. They can even be run on multiple computers with different configurations.

1.6 How Automation Script Works



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

1.7 Automation Testing Process



1. **Test tool selection:** Test Tool selection largely depends on the technology the Application Under Test is built on. For instance, we want to perform web application testing, we will choose the tool which works good with the web applications, like we can use selenium for web application.
2. **Define the scope of Automation:** The scope of automation is the area of your Application Under Test which will be automated. Following points help determine scope:
    a. The features that are important for the business
    b. Scenarios which have a large amount of data
    c. Common functionalities across applications
    d. Technical feasibility
    e. The extent to which business components are reused
    f. The complexity of test cases. Ability to use the same test cases for cross-browser testing
3. **Planning, Design, and Development:** During this phase, you create an Automation strategy & plan, which contains the following details-
    a. Automation tools selection
    b. Framework design and its features
    c. In-Scope and Out-of-scope items of automation
    d. Automation test script preparation
    e. Schedule and Timeline of scripting and execution
    f. Deliverables of Automation Testing
4. **Test Execution:** Automation Scripts are executed during this phase. The scripts need input test data before there are set to run. Once executed they provide detailed test reports. Execution can be performed using the automation tool directly or through the Test Management tool which will invoke the automation tools. Scripts can be executed in a single machine or a group of machines. The execution can be done during the night, to save time.
5. **Maintenance:** As new functionalities are added to the System Under Test with successive cycles, Automation Scripts need to be added, reviewed and maintained for each release cycle. Maintenance becomes necessary to improve the effectiveness of Automation Scripts.

1.8 Why Framework for Automation

- Maintaining consistency of Testing
- Improves test structuring
- Minimum usage of code
- Less Maintenance of code
- Improve re-usability
- Non-Technical testers can be involved in code
- The training period of using the tool can be reduced
- Involves Data wherever appropriate

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

II. TYPES OF AUTOMATION FRAMEWORK

1. **Data Driven Automation Framework:** Data-driven test automation framework is focused on separating the test scripts logic and the test data from each other. Allows us to create test automation scripts by passing different sets of test data. The test data set is kept in the external files or resources such as MS Excel Sheets, MS Access Tables, SQL Database, XML files etc., The test scripts connect to the external resources to get the test data. By using this framework we could easily make the test scripts work properly for different sets of test data. This framework significantly reduces the number of test scripts compared to module-based framework.
2. **Keyword Driven Automation Framework:** The Keyword driven testing framework is an extension to Data driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file. These set of code are known as Keywords and hence the framework is so named. Keywords are self-guiding as to what actions need to be performed on the application. The keywords and the test data are stored in a tabular like structure and thus it is also popularly regarded as Table driven Framework. Take a notice that keywords and test data are entities independent of the automation tool being used.
3. **Modular Automation Framework:** In the modular testing framework, testers create test scripts on module wise by breaking down the complete application under test into smaller, independent tests. In simple words, testers divide the application into multiple modules and create test scripts individually. These individual test scripts can be combined to make larger test scripts by using a master script to achieve the required scenarios. This master script is used to invoke the individual modules to run end to end test scenarios.
4. **Hybrid Automation Framework:** Hybrid Test automation framework is the combination of two or more frameworks mentioned above. It attempts to leverage the strengths and benefits of other frameworks for the particular test environment it manages. Most of the teams are building this hybrid driven framework in the current market.

2.1 Automation at Different Levels

- **The level of unit testing (Unit Test layer) –** at this level automated tests are the component or unit tests designed by developers. Testers can also write such tests, if they have the necessary skills. Similar tests in the early stages of the project, as well as their constant actualization and the addition by new tests, checking the “bug fixes”, can help to protect the project from the many serious problems.
- **The level of functional testing (Functional Test Layer non-UI) –** unfortunately, not all the application’s business logic can be tested using the GUI layer. This could be a feature of implementation, which hides the business logic from the user. Access directly to the functional layer, which gives the opportunity to test the business logic of the application directly bypassing the user interface, is provided for the testing team only in agreement with the developers.
- **The level of the user interface testing (GUI Test Layer) –** at this level, it is possible to test not only the user interface, but also the functionality by performing the operation causing the business logic of the application. This kind of cross-cutting tests provides a greater effect than just testing the functional layer, as it is necessary to perform functional testing by emulating the end-user operations through a graphical interface.

2.2 Automation Testing tools

A. Selenium

- It is a software testing tool used for Regression Testing. It is an open-source testing tool that provides playback and recording facility for Regression Testing. The Selenium IDE only supports Mozilla Firefox web browser.
- It provides the provision to export recorded script in other languages like Java, Ruby, RSpec, Python, C#, etc
- It can be used with frameworks like JUnit and TestNG
- Autocomplete for Selenium commands that are common
- Walkthrough tests
- Identifies the element using id, name, X-path, etc.
- Store tests as Ruby Script, HTML, and any other format

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

- It provides an option to assert the title for every page
- It allows to insert comments in the middle of the script for better understanding and debugging

B. QTP (Microfocus UFT)

QTP is widely used for functional and regression testing, it addresses every major software application and environment. To simplify test creation and maintenance, it uses the concept of keyword driven testing. It allows the tester to build test cases directly from the application.

It is easier to use for a non-technical person to adapt to and create working test cases

It fix defects faster by thoroughly documenting and replicating defects for developer

Collapse test creation and test documentation at a single site

Parameterization is easy than WinRunner

QTP supports .NET development environment

It has better object identification mechanism

It can enhance existing QTP scripts without "Application Under Test" is available, by using the active screen

C. JMeter

JMeter is used for load testing and measure performance. You can use JMeter to analyze and measure the performance of web application or a variety of services. Performance Testing means testing a web application against heavy load, multiple and concurrent user traffic. JMeter originally is used for testing Web Application or FTP application.

JMeter is an Open-Source testing software. It is 100% pure Java application for load and performance testing.

JMeter is designed to cover various categories of tests such as load testing, functional testing, performance testing, regression testing, etc., and it requires JDK 5 or higher.

This tutorial provides an in-depth coverage of JMeter framework including its test plans, listeners, functions, and regular expressions.

2.3 Agile in Software Testing

A. What is Agile?

In simple words, agile is marked by ready ability to move with quick easy grace. It means having a quick, resourceful and adaptable character.

B. What Does That Mean?

- Process has to be lightweight and sufficient
- Lightweight helps us adapt and move
- Sufficient recognizes our ineffectiveness to be complete and relies on strong communication

C. Background of Agile

- Agile isn't a set of tools or a single methodology, but a philosophy
- It was put to paper in 2001 with an initial 17 signatories. Agile was a significant departure from the heavyweight document-driven software development methodologies—such as waterfall—in general use at the time.

D. Manifesto for Agile

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

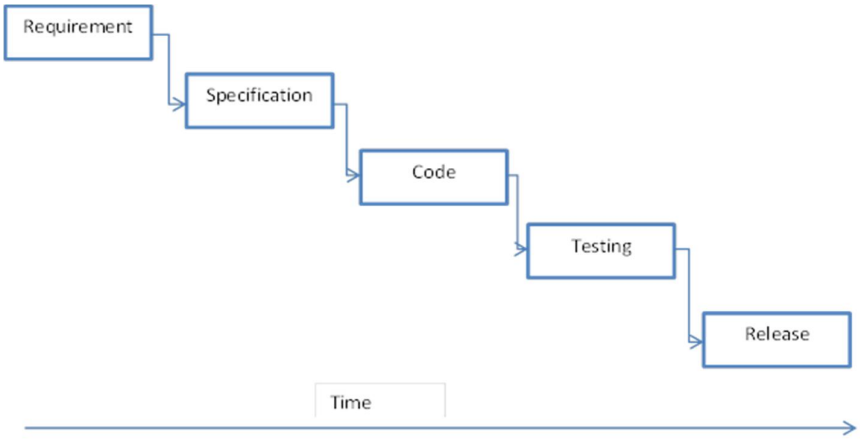
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

How is Agile testing Different? Traditional vs. Agile

In the phased approach as shown in the diagram, it is clear that testing happens at the end, right before release. The diagram is idealistic; because it gives the impression there is as much time for testing as there is for coding. In many projects, this is not the case. The testing gets "squished" because coding takes longer than expected, and because teams get into a code-and-fix cycle at the end.

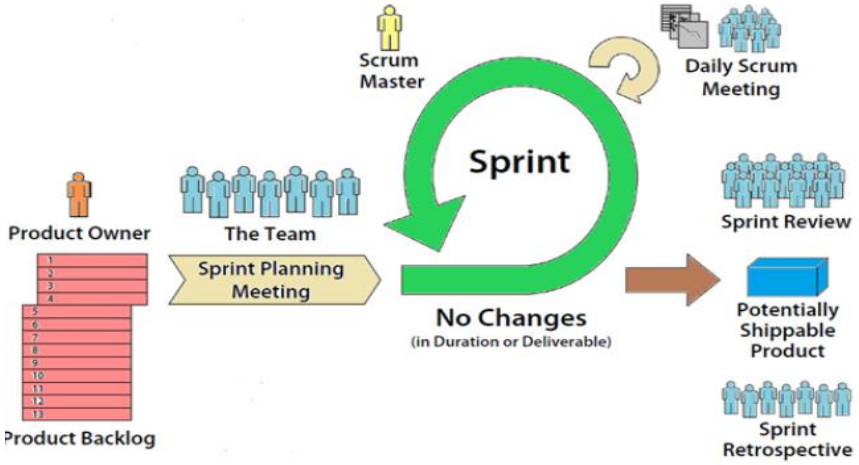


Agile is iterative and incremental. This means that the testers test each increment of coding as soon as it is finished. Iteration might be as short as one week, or as long as a month. The team builds and tests a little bit of code, making sure it works correctly, and then moves on to next piece that needs to be built. Programmers never get ahead of the testers, because a story is not "done" until it has been tested.

E. Agile Terminologies

- Sprint – The time frame in which the work must be completed – often 30 days.
- Daily scrum – Lead by the scrum master, the team comes together for short daily meetings, in which they discuss what they have completed, what they are working on and any issues that are blocking the work
- Sprint end - At the end of a sprint, two meetings are held:
- Sprint review – The team shows their work to the product owner.
- Sprint retrospective – The team discusses what they can do to improve processes. An important goal is continuous improvement.

F. Agile – Scrum



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

III. TEST CASE DESIGN

3.1 Test Analysis and Design

Test Analysis and Design is a Fundamental Test Process which creating test conditions & test cases. In this process performed major tasks like reviewing the test basis, identifying the test conditions based on analysis of test items, writing test cases, identifying necessary test data to support the test conditions and test cases, designing the test environment.

3.2 What is a Test Case?

A Test Case refers to the sequence of actions required to verify a specific feature or functionality. Essentially, the test case details the steps, data, prerequisites, and postconditions necessary to verify a feature.

3.3 The Objective of Writing Test Cases

- To validate specific features and functions of the software.
- To record a catalog of steps undertaken, which can be revisited in the event of a bug popping up.
- To help detect usability issues and design gaps early on.
- To help new testers and devs quickly pick up testing, even if they join in the middle of an ongoing project.

3.4 Test case development

- **Test Case ID –** Unique ID is required for each test case.
- **Test Scenario –** module / feature/functionality that can be tested
- **Prerequisites –** Any prerequisite that must be fulfilled before the execution of this test case.
- **Test Steps –** series of steps that will be followed while testing the software module's
- **Test Data –** is the input given to a software program during test execution

3.5 Test Case Development

- **Expected Result –** What should be the outcome expected from system when these steps are followed.
- **Actual result –** What is actual result generated as a result of following the test case steps
- **Test Status –** Pass/Fail – If an actual result is not as per the expected result, then mark this test as failed. Otherwise, update it as passed

3.6 Best Practices for Writing Test Cases

- Prioritize clarity and transparency. Be clear, concise, and assertive in describing.
- Focus on End-User requirements when writing sample test cases. Map test cases to reflect every aspect of the user journey. Use the Specifications Document and the Requirements Document to do so.
- Avoid repetition. If multiple tests can be executed with the same test case, use the Test Case ID to refer to the required test case.
- Keep Test Steps as minimal as possible. Ideally, to 10-15 steps
- Focus on achieving maximum test coverage.

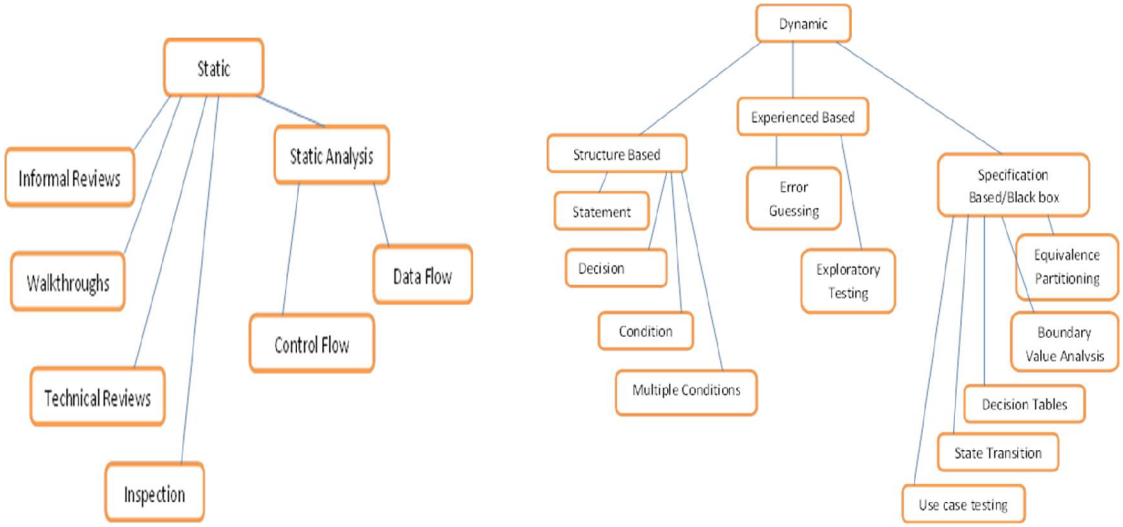
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

3.7 Test Case Writing Practice with Example

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check Customer Login with valid Data	1. Go to site http://demo.guru99.com 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = guru99 Password = pass99	User should Login into an application	As Expected	Pass
TU02	Check Customer Login with invalid Data	1. Go to site http://demo.guru99.com 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = guru99 Password = glass99	User should not Login into an application	As Expected	Pass

3.8 Test Design Techniques

Test design technique is a procedure for determining test conditions, test cases and test data during software testing. There are many different types of software testing technique, each with its own strengths.



3.9 What is Static Testing?

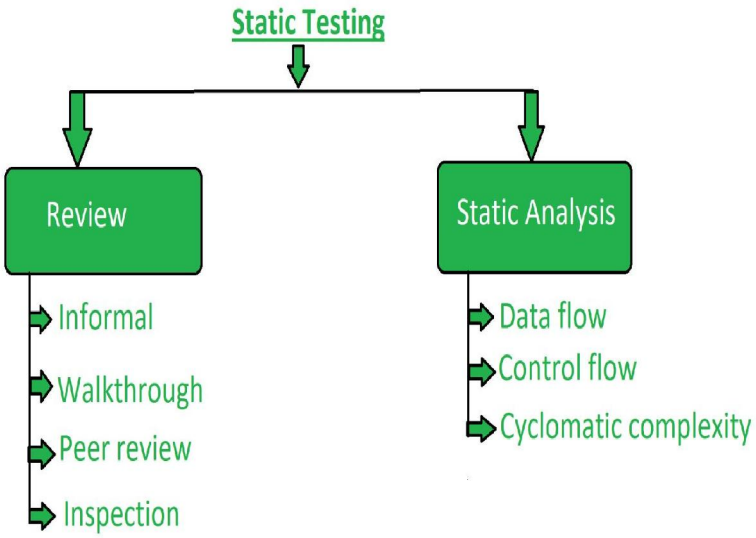
Static Testing, a software testing technique in which the software is tested without executing.
Instead of executing the code, static testing is a process of checking the code and designing documents and requirements before it's run in order to find errors. The main goal is to find flaws in the early stages of development. It is normally easier to find the sources of possible failures this way.

3.10 Static Testing Technique

Static testing is carried out with two different steps or techniques:

- **Review** - Typically used to find and eliminate errors or ambiguities in documents such as requirements, design, test cases, etc.
- **Static analysis** - The code written by developers are analyzed (usually by tools) for structural defects that may lead to defects.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252



3.11 Types of Static Testing Reviews

This process can be carried out in four different ways:

- **Informal --** informal reviews will not follow any specific process to find errors. Coworkers can review documents and provide informal comments.
- **Walkthrough --** the author of whichever document is being reviewed will explain the document to their team. Participants will ask questions, and any notes are written down.
- **Inspection --** a designated moderator will conduct a strict review as a process to find defects.
- **Technical/peer reviews --** technical specifications are reviewed by peers in order to detect any errors.

3.12 Types of 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.

Static Analysis is of three types:

- **Data Flow**: Data Flow Testing is a specific strategy of software testing that focuses on data variables and their values
- **Control Flow**: Control flow is basically how the statements or instructions are executed.
- **Cyclomatic Complexity**: Cyclomatic complexity is the measurement of the complexity of the program that is basically related to the number of independent paths in the control flow graph of the program.

For example, if source code contains no control flow statement then its cyclomatic complexity will be 1 and source code contains a single path in it. Similarly, if the source code contains one if condition then cyclomatic complexity will be 2 because there will be two paths one for true and the other for false.

3.13 Dynamic Testing Technique

Dynamic Testing is a software testing method used to test the dynamic behaviour of software code. The main purpose of dynamic testing is to test software behaviour with dynamic variables or variables which are not constant and finding weak areas in software runtime environment.

Dynamic techniques are subdivided into three more categories:

- Black box testing techniques (Also known as specification-or behavioral techniques)
- White-box testing techniques (also known as structure-based or structural techniques) and
- Experience-based techniques

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, June 2022
Impact Factor: 6.252

Black Box Testing Technique

This testing technique exercises a system end-to-end. Just like end-users "don't care" how a system is coded or architected and expect to receive an appropriate response to their requests, a tester can simulate user activity and see if the system delivers on its promises. Input test data is given to the system, which is a black box to the tester, and results are checked against outputs after executing the software.

- Equivalence Partitioning
- Boundary Value Analysis
- Decision Table Testing
- State Transition Testing
- Use case Testing

BIBILOGRAPHY

[1]. https://www.guru99.com/software-testing-introduction-importance.html
[2]. https://smartbear.com/learn/automated-testing/test-automation-frameworks/
[3]. https://www.guru99.com/agile-testing-a-beginner-s-guide.html
[4]. https://reqtest.com/testing-blog/test-case-design-techniques/