

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management



Diploma Thesis

Software Quality Assurance and Testing Processes

Ankit Vaniya

©Year of submission 2021-2022 CULS

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

B.Sc. Bc. Ankit Vaniya

Systems Engineering and Informatics
Informatics

Thesis title

Software Quality Assurance and Testing Processes

Objectives of thesis

The thesis software quality assurance and testing processes is focused on the software quality assurance processes and software testing approaches. This thesis main aim is to provide the detailed work flow of the software testing levels as well as the detailed information about the software quality assurance process, quality assurance plan, tasks, processes and difference between software quality assurance and software testing process are carried out during the process.

Methodology

This thesis includes the methodology about the software development cycle.

This thesis is more focused on the Quality assurance and Quality control process. Also focused on the testing types and their detailed description. Moreover it will include some practical work with testing and quality control documents like test cases, test scripts, Test scenarios, bug report, test evaluation report, test plan and test strategy.

The proposed extent of the thesis

60 – 80 Pages

Keywords

Software assurance, Automation, Selenium, software testing

Recommended information sources

Brian Marick, The Craft of Software Testing, Prentice Hall, 1995. ISBN-13: 978-0131774117

“Software Negligence & Testing Coverage, ISBN-13: 978-1138468429

Software Testing: Principles and Practices, ISBN-13: 978-8177581218

Expected date of thesis defence

2021/22 SS – FEM

The Diploma Thesis Supervisor

prof. Ing. Mansoor Maitah, Ph.D. et Ph.D.

Supervising department

Department of Economics

Electronic approval: 25. 3. 2022

prof. Ing. Miroslav Svatoš,
CSc.

Head of department

Electronic approval: 28. 3. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Dean

Prague on 28. 03. 2022

Declaration

I declare that I have worked on my diploma thesis titled “Software Quality Assurance and Testing Processes” by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on **25 March 2022**

Ankit Vaniya

Acknowledgement

I declare that the thesis “Software Quality Assurance and Testing Processes” is developed independently under the guidance of my supervisor, Prof. Ing Mansoor Maitah, PH.D for which I am thankful to him and other persons specially my parents and my brother for their advice and support given during my research. I also declare that the use of literature and other information sources that are cited in the work are listed in the reference at the end of work. As the author of the referred thesis, I further declare that I didn't infringe the copyright of any third parties in the context of its creation.

Shrnutí

V této diplomové práci jsou zpracovány podrobné informace o procesech zajišťování kvality softwaru a testování softwaru. Tato práce podrobně vysvětluje základy procesů zabezpečování jakosti a úkoly, které jsou prováděny v průběhu celých procesů zabezpečování jakosti. Kromě toho jsem se zmínil o dokumentaci plánu zajištění kvality a o tom, které nezbytné informace by měl provádět, a také jsem vysvětlil, jak se procesy zajištění kvality softwaru a testování softwaru liší.

Tato práce je také zaměřena na testování softwarové automatizace. Vysvětluje celý proces testování automatizace softwaru a rozdíl mezi manuálním a automatizačním testováním. Dále jsou jako příklad popsány informace o rámcích automatizačních testů, typu rámců a o testování výkonu. Navíc obsahuje vysvětlení procesu všech úrovní testování v podrobném a praktickém příkladu na každé úrovni testování v pořadí.

Klíčová slova: Zajištění kvality softwaru, testování automatizace softwaru, ruční testování, rámec pro testování automatizace, testování výkonu, úroveň testu jednotek, úroveň testu integrace, úroveň systému, úroveň akceptace, nástroj pro automatizaci selenu.

Summary

This thesis is conducted the detailed information about the software quality assurance and software testing processes. This thesis deeply explained about the basic of quality assurance processes and the task which are carried out during the entire quality assurance processes. Moreover, I mentioned about the quality assurance plan documentation and which necessary information it should conduct as well as it explained how software quality assurance and software testing processes are different.

This thesis is also focused on the software automation testing. It is explaining about the entire software automation testing process and the difference between the manual and automation testing. Additionally, it is described the information about the automation test frameworks, type of framework and about the performance testing as an example. Moreover, It is containing the all the testing levels process explanation in detailed and practical example on each testing level in sequence.

Keywords: Software quality assurance, Software automation testing, manual testing, automation testing framework, performance testing, unit test level, integration test level, system level, acceptance level, selenium automation tool.

Table of contents:

Table of Contents

1. Introduction.....	1
1.1 What is software quality assurance?	1
1.2 Importance of software quality assurance	2
1.3 What is software testing.	3
1.4 Importance of software testing	5
1.5 What is automation testing	5
1.6 Importance of automation testing	7
2. Goal of Thesis	8
3. Methodology of Thesis	9
4. Quality assurance	10
4.1 Basics of quality assurance	10
4.2 The quality assurance plan	10
4.3 Software quality assurance task	12
4.4 Difference between Software quality assurance and software testing	16
5. Automation testing.....	17
5.1 Basics of Automation testing	17
5.2 Automation testing process	18
6. Automation test framework	21
6.1 Basics of Automation test framework	21
6.2 Types of frameworks used in Automation testing	21
6.3 Difference between manual testing and automation testing	29
7. Performance testing	31
7.1 Introduction to performance testing	31
7.2 Measurements used for performance metrics	32
7.3 Performance testing processes	34
7.4 Good practices and recommendations for the performance test	36
7.5 Types of performance testing.....	38
8. Software testing levels	42
8.1 Introductions to software testing levels	42

8.2 Unit testing	43
8.2.1 Black-box testing.....	44
8.2.2 White-box testing	44
8.2.3 Gray-box testing	45
8.3 Integration testing.....	45
8.3.1 Big-bang approach.....	47
8.3.2 Incremental testing	48
8.3.3 Top-down approach.....	49
8.3.4 Bottom-up approach	50
8.4 System testing	51
8.4.1 Testing types.....	53
8.5 Acceptance testing	61
8.5.1 Types of acceptance testing.....	63
8.5.2 Acceptance testing process workflow	66
8.5.3 Guidelines for the acceptance testing	68
9. Practical Part	70
9.1 Unit testing	70
9.1.1 Test case sample for Unit testing.....	70
9.1.2 Screenshots of Unit testing.....	71
9.1.3 Source codes for Unit testing	72
9.2 Integration testing.....	74
9.2.1 Test case sample for Integration testing	74
9.2.2 Screenshots of Integration testing	75
9.2.3 Source codes for Integration testing	76
9.3 System testing	77
9.3.1 Test case sample for System testing.....	77
9.3.2 Screenshots of System testing	79
9.3.3 Source codes for System testing.....	79
9.4 Test case execution video link	80
9.5 Approximate data related the time/cost/factors of manual and automation testing	80
9.5.1 Approximate time taken for Test case/ Script preparation.....	80
9.5.2 Manual tester's approximate test case execution time with hourly/monthly wages	81
9.5.3 Automation tester's approximate test case execution time with hourly/monthly wages	81

9.5.4 Comparison of manual and automation testing as per the software testing factors	82
9.6 Economic benefits of Software quality assurance, Manual testing and Automation testing	82
9.7 Why selenium tool is good for Automation testing?	84
10. Conclusion	87
11. Bibliography	88

Table of Figures:

No of Figure	Description	Page no.
1.1.1	Software quality assurance	1
1.3.1	Software development life cycle	3
1.3.2	Software testing life cycle	4
1.5.1	Benefits of automation testing	6
6.2.1	Data Driven Automation Framework	22
6.2.2	Keyword Driven Automation Framework	23
6.2.3	Hybrid Automation Framework	24
6.2.4	Module-based Test Framework	25
6.2.5	Component of automation testing framework	27
6.2.6	Library architecture Test Framework	27
7.3.1	Performance testing processes	34
7.5.1	Types of Performance testing	38
8.1.1	Software testing levels	42
8.2.1	Unit testing	43
8.3.1	Integration testing	46
8.3.1.1	Big-bang approach	47
8.3.2.1	Incremental testing	48
8.3.3.1	Top-down approach	49
8.3.3.2	Top-down approach	50
8.3.4..1	Bottom-up approach:	50
8.4.1	System testing	52
8.4.1.1	Testing types	53
8.5.1	Acceptance testing	62
8.5.1.1	Types of acceptance testing	63
8.5.2.1	Acceptance testing process workflow	66
9.1.2.1	Screenshots of the Unit-testing	71
9.1.2.2	Screenshots of the Unit-testing	72
9.1.2.3	Screenshots of the Unit-testing	72
9.2.2.1	Screenshots of the Integration-testing	75
9.2.2.2	Screenshots of the Integration-testing	76
9.3.2.1	Screenshots of the System-testing	79
9.7.1	Comparison of selenium with other automation tools	86

Tables:

No of Table	Description	Page no.
4.4	Difference between Software Quality Assurance and Software Testing:	16
6.3	Difference between manual testing and automation testing:	29
9.5.1	Approximate time taken for Test case/ Script preparation	80
9.5.2	Manual tester's estimated test case execution time with hourly/monthly wages	81
9.5.3	Automation tester's estimated test case execution time with hourly/monthly wages	81
9.5.4	Comparison of manual and automation testing as per the software testing factors	82
9.6	Economic benefits of Software quality assurance, Manual testing and Automation testing	82
9.7	Selenium compatibility with language, platform and browsers.	85

1. Introduction

1.1 What is software quality assurance?

The quality assurance is type of software testing process which is gives a certification or positive declaration on the quality of software service or product provided by the individual organization. The software quality assurance is process which verify and validate the software product or service according to the defined software engineering standards, processes, methodologies, and quality testing activities. Its main goal to provide the guaranty of quality that prove the organization is delivering the best and error free software product which helps organization to gain customer trust and satisfaction. The software quality assurance processes involve in software development from the gathering re-equipment to release the software product to the market to improve and ensure the quality. (Dyer, 1992).

Figure-1.1.1



(Google Images, n.d.)

The software quality assurance process is not a single activity process. It is containing the activities like setting the checkpoints, measuring the change impacts, have multiple testing strategy and approaches, maintaining the records and reports, managing the relations, quality assurance test plans, etc. The software quality assurance includes so many techniques in process like auditing, reviewing, design inspections, standardization, static analysis, code inspections, functional, integration and interface inspection, documentation of records and reports and about the risk analysis. [_\(Marick, 1995\).](#)

1.2 Importance of software quality assurance:

The majority of software provider are following the software quality assurance process to deliver the best quality of software product or service. The software quality assurance process is involved from the beginning, so it is maintaining the software engineering standards and processes so it help to prevent the company image in market. In other words, it is helping the organization to maintain it good while in this competition market and help to attract more customer to make profitable business.

Meanwhile, Software quality assurance process proven time saving and cost-effective process. It helps to identify the bug in early stages in complex project so in the end it helps to save time and preventing the software product from the unwanted potential errors. Nowadays, security is becoming a key factor in the software product market. Software quality assurance helps to detect the weakness of the software product which will uncover the vulnerabilities and loopholes in software product and make secure and trustworthy software product for the people. So like this several ways software assurance provide positive declaration to people and guaranty them about the quality of software service or product. [_\(Marick, 1995\).](#)

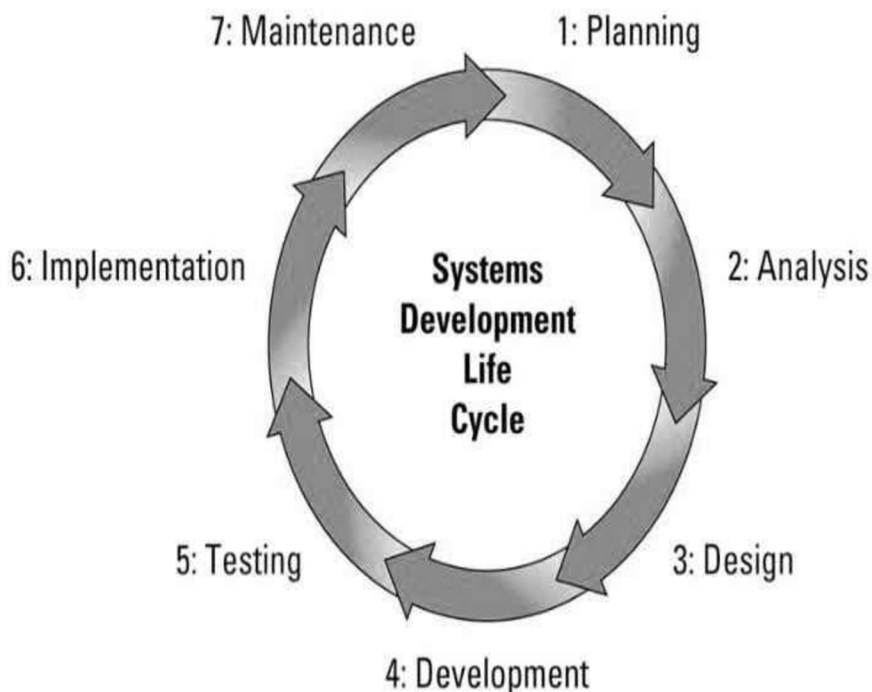
1.3 What is software testing?

The software testing can be defined in various ways. I can say that the software testing is process of verification and validation process to evaluating the software product quality and performance. In other words, software testing is technique to evaluating the software product's component and its functionality that it is working as per the specified requirements as the customer needs.

Validation: Validation process is performed to ensure that the software product is meeting and satisfying the specific requirement which is defined by the client or customer. The validating is processed to check that the software product is developed as per the specific requirements.

Verification: Verification process is performed to ensure that the software product is satisfying the requirements imposed as the development part. In other words, the software product supposed to behave as we want it to behave. [_\(Vaniya, 2018\).](#)

Figure-1.3.1



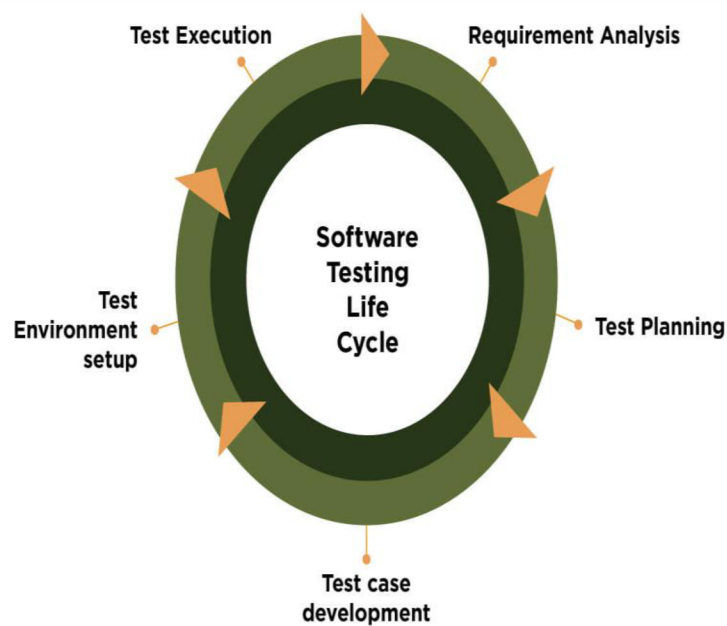
[_\(Google Images, n.d.\)](#)

The software testing is not a single step process. It is containing the several stages to complete the whole testing process. By the way software testing is also one of the stages of software development life cycle. Software testing start from to gathering the requirement. Ones the requirement gathering, and testing is done by the business analyst. It broke into the different part and move it to the planning stage for testing process.

The planning is the initial stage where they have plan whole testing process, testing aspect, scopes, and documentation. Meanwhile they are also working for preparation on test analysis and test process design. In this stage they will create a properly well-defined test plan and test strategy. After the test plan and strategy document available tester starts working to prepare the test case. Then the process moves to the implementation and execution where test perform the several testing types on the software product.

After the test run completed, they collect the test result and document it in the system and starts process for the test exit criteria and later they sign off the test activity. (Marick, 1995).

Figure-1.3.2



Jelvix

Source: App Hawks

jelvix.com

_(Software-testing-life-cycle, n.d.)

1.4 Importance of software testing:

Software testing one of the most important parts of the software development life cycle. It is playing the vital role in the whole software development process in several ways. Software testing is most important as it is verifying and validating the software product as per the given requirement form the customer.

Its help us gain and satisfy the customer from the quality of the software product which can be ensure by the software testing. Nowadays people are only the websites and software product which are secure enough, so Software testing help to identify security related bug. If software testing process starts form the early stage of the software development, it can help to find and correct a bug in early stage so it can help to reduce the cost of bug fixing.

The software testing process help to reduce the loopholes in the software product which can help to enhance and improve the software development process. [_\(Vaniya, 2018\).](#)

1.5 What is Automation testing?

Automation testing is a type of software testing which is executing by the automation tool. Automation tester prepared the necessary coding or script into automation tool to performed test step and test data and executing the test case by tool itself without any human efforts. [_\(Kaner, 1996\).](#)

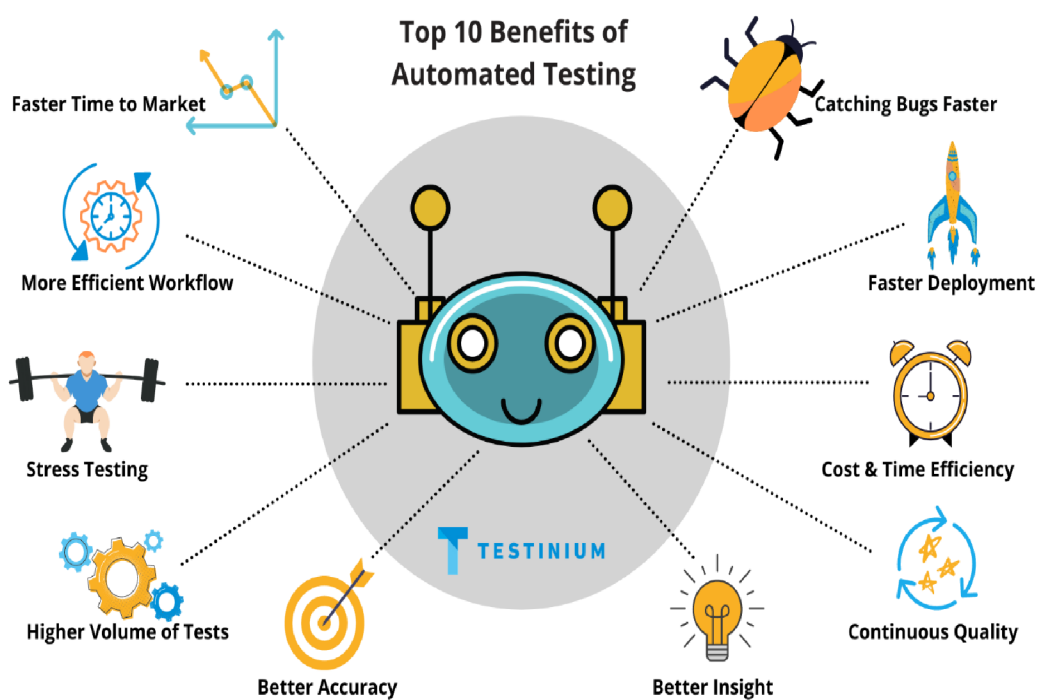
Automation tool help to run the same test several time and record the test result in selected file format and deliver the test result more accurate than human. It is time saving process the manual testing. The automation testing is most use full to perform the non-function testing process which are hard to conduct by the human efforts as well as it is suitable for the functional testing process also. Main aim of automation testing in software development life cycle which is about to save time and money.

The software development is huge process with its several stages like planning, analysis, design, development, testing, implementation, and maintenance. During the time of development process many times happens that there are changes in requirement, detecting bug, fixing bugs, new feature added or so many unwanted situations might come in middle of development project. Nowadays majority of

development team starts the testing process in early stage to avoid major loss to time and money and try to fix the defect in early stages.

But sometime because of changes or fixes tester need to retest all the test case from start to end to verify that because changes in requirements or fixes the bug in some corner of software product is not affecting or creating the problem in other part of software product. In this kind of situation test solution for the tester to execute the testing process by the automation tool. [\(Kaner, 1996\)](#).

Figure-1.5.1



[\(Top-10-benefits-of-automated-testing, n.d.\)](#)

1.6 Importance of automation testing:

There are so many benefits of the automation testing as we can see in the figure above but few more like its help to developer and tester to understand better about the code mechanism and root of bug. It helps tester to explain in better to developer about the bug priority and severity.

When tester executing the same test cases again with automation tool it gives them more challenges and improve their skills. As well as same automation test framework of created test case they can use is more than one project.

As I already mentioned that the automation testing's main aim to do the task which is hard to done by the human like creating virtual users to test software product like web application when it comes to evaluation it's performance, scalability, stability, security, and network related non-functional aspects. As well as for this kind of non-functional testing aspect need accurate test report to deliver the best quality of the software product. [\(AutomatedTestingHandbook, n.d.\)](#)

2. Goal of thesis

The thesis software quality assurance and testing processes is focused on the software quality assurance processes and software testing approaches.

This thesis contains the detailed information about the software quality assurance process, quality assurance plan, tasks, processes and difference between software quality assurance and software testing. As well as about the software testing process type like manual testing and automation testing.

Also, this thesis contains the information about the automation testing and its approaches. Moreover, it includes the detailed information about the software testing levels.

This thesis also includes some practical examples which will help to clear the concept of automation testing and how its work in automation tool.

3. Methodology of thesis

The software quality assurance and testing process thesis includes the methodology of software quality assurance and software testing process as per the all the necessary software development engineering standards. The thesis main goal is to focus on quality verification and validation processes according to testing types like manual testing and automation testing.

Moreover, it gives basic idea about the black box, white box and gray box testing. It also includes functional and non-functional testing. The thesis provides details information about the testing levels which are unite test, integration test, system test and acceptance test.

Furthermore, this thesis includes the explanation on the automation testing and focus on how automation testing is prepared, design and work. Also, this thesis includes the practical example on manual and automation test using selenium automation tool and testing documentation.

4. Quality assurance

4.1 Basics of quality assurance:

Quality assurance is the process to verify the quality of the software product or the software service provided by the particular organization to the relevant customer. Quality assurance is process which mainly focus to maintain and improve process by effective and efficient way. The set of process and the standards decided by the QA team will help to control the product defects before the launch in market. In our case of software development and testing field the quality assurance is mainly focused on the software development process and testing process.

The quality assurance is also knowing as QA testing which mainly verify the quality of the software product according to the defined quality standards for the software product or software service. The QA processes to verify the quality of product could be different according to product and service provided by the organization. The QA process are best practice to maintain a good relationship between client and organization. It will help organization to gain trust and loyalty of client. The QA process prime goal is to ensure the quality of product or service. [\(Dyer, 1992\).](#)

4.2 The quality assurance plan:

The software quality assurance plan mainly includes the description of QA processes, Areas to review and control, documentation about the product with step-by-step instructions, QA techniques, responsibilities to team, and environment tools to check the and ensure the quality to product.

The QA plan documentation includes the detailed information about the different section which is mainly depends on the product or service provided by the origination. E.g. the document's sections can be as per below. [\(assurance, n.d.\)](#)

a) Purpose section:

This section should contain the information regarding the main goal of the QA plan. Also, we can say this section can be stated as the detailed description of the QA plan which will be focus on the whole QA process.

b) Reference:

This part should contain the information about the reference, other useful documents or contact person information.

c) Software configuration management:

This part should have detailed information regarding the product or service configurations files. Which will be helpful to check the quality of product in different technical environment.

d) Technical environment, tools, and methodologies:

This part should have detailed information about the tool using to report the bugs. The tools will give the exact statistical information about the bug and progress report. We can easily encounter the problem and know their status and prepare a good execution plan for repair strategy. Also the tools will be helpful for estimate the project progress.

The technical environment information most use full because we need to cover our product performance and behaviour according the technical environment. The software product might be behave different according to technical environment.

e) Bug reporting and repair action:

This section will provide the information regarding the progress of the bug the prepared the step for the repair actions. Also it include the information regarding the work flows so it can helpful for estimate the work according to bug status.

f) Code control:

This section is assuring that all modules code are tested and successfully working as per the specific requirement.

g) Documentation, Collection, Maintenance and retention :

This section contains the all the detailed documentations like collection report, retention report and maintenance report.

h) Testing methodology:

This section includes the information regarding the methodology needs to perform during the QA processes. Testing methodology might be different according to the organization as well as the software product provided.

It might also include the information about testing techniques, testing types and estimation methods.

4.3 Software quality assurance task:

The software quality assurance tasks are mainly includes the following sections. ([AutomatedTestingHandbook, n.d.](#))

A. Create a software quality assurance plan:

The SQA plan mainly focus on the how and when need to check the quality of the software product. It's mainly included the different quality check techniques and software engineering approaches.

Moreover, it contains the information about the team provided and their technical skills. This is the most important and necessary document to carry out the software quality assurance processes. [_\(Dyer, 1992\).](#)

B. Set the key feature to check the quality at different development stage:

This activity is majority planned by the QA engineers in the organization. They decide the key check point at different stages of development life cycle. They make schedule for it and ensure the quality check and documented it according to the different stage of software development. Also mentioned the software features are check for quality inspection as per the schedule.

C. Software quality assurance techniques:

This activity is about the SQA techniques are to apply on the software product on different stages of development to achieve the high quality of inspection.

D. Multi testing strategy:

The multi testing strategy will be appropriate approach to ensure the better quality of software product. It means that using multi testing type will have better result instead rely on a single testing approach at all stage of development.

E. Previous technical review:

The previous technical review will help to find out the bugs in the early phases of the development life cycle and reduce the work of bug fixing processes. In the process they conduct a meeting of the technical staff and discuss about the actual quality requirement and with the use of that will design and prepare a proper quality of prototypes.

F. Enforcing factor of process:

The process of the development and quality assurance should be stick on the defined procedures. This activity insists to follow the defined processes during every stage of software life cycle. [\(Dyer, 1992\)](#).

This activity contains two sub activities as per below

- **Product evaluation**

This activity confirms that the software product is match with the requirements mentioned in the project management plan. This also confirms that the defined standards are followed in the correctly in the software development process.

- **Process monitoring**

This activity verifies that the proper step and standards followed during the software development process. This is done by the matching the followed steps against the mentioned steps in the project management plan.

G. Controlling change:

This process is done after the bug fixed. In this activity usually done by manually or automated tool to ensure the change after the bug fixed.

This activity validates the change request and controlling the behaviour of the software product after the request has been don and make sure that the quality is maintained during the development, testing and maintenance stages.

H. Change impact measurement:

Whenever the tester or QA engineer find the bug it is assigned to the responsible developer to correct it. After that it need to retest the bug to confirm that it is correctly fixed as well as we need to also verify that the change we had just made to fix the bug it has not affected to the other part of software product.

([AutomatedTestingHandbook](#), n.d.)

This kind of testing process is called the regression testing. For this kind of activity manager and developers records in quality matrix so they can get idea and take appropriate action in the process of software development at any phase of development.

I. Quality assurance audit:

The quality assurance audit verifies the whole process of the software development as per the project management documents. This process depend on the reports provided by the team with all process status and record and compared with the actual plan to verify that the all process are actually taken in consideration while the development process.

J. Maintaining records:

It is really very important and good practice to maintain the records, reports, test result and the software product project documentations.

It will be always useful to get review for future reference. This documents will also useful for the project presentation to the non-technical audience like management body or stakeholders.

K. Collaboration in team:

It is very important to have a good collaboration between team during the software development. We all know that there are two main team developer and tester that need to have good understanding and give equal importance to individual thinking. We often see that there is understanding issue between tester and developers and it can really affect to the process growth. It should be avoided as much as possible and keep good collaboration between team is most important. [\(Dyer, 1992\)](#).

4.4 Difference between Software Quality Assurance and Software Testing:

No	Software Quality Assurance	Software Testing
1	Software quality assurance process ensure the implementation of define process and standards in developed software product as it is documented in project management plan.	Software testing is process to identify the bug in the developed software product.
2	SQA is process focused activity.	Software testing is product focused activity.
3	SQA processes are defined to applies on all products will be created by the company.	Software testing process are defined to apply on the particular product being tested.
4	SQA is preventing type of activity.	Software testing is corrective type of activity.
5	SQA process main aim is to assure the quality of software product.	Software testing process main aim is to control the quality of software product.
6	SQA processes are mainly focused on assure the quality as well as meeting the requirements.	Software testing mainly focused to test the product on unit level, integration level, system integration level.
7	The whole project team is involved in SQA.	Only the testing team is involved in software testing.
8	It is done by the human checking of documents and records.	It is manual done by the human by clicking the in computer according to testing steps or by the automated software.
9	Proactive measure.	Reactive measure.
10	SQA ensure that the right things are being done by the project team.	Software testing team evaluate the results of the done things

5. Automation testing

5.1 Basics of Automation testing:

The software automation testing is defined as the process of software testing which is done by the automation tool. The automation testing is basically created to reduce the human efforts from the repetitive tests. Usually while in software development life cycle we are used to test as per the development phases are done. The process of testing is different in the different model of the software development life cycle. The automation testing is mainly come to in concept because of fixing or changes needed whenever requirement are changed, or some error or bugs found. After the fixes or changes we needed to execute all the test cases which are already executed just to make sure there is not any problems occurs because of fixing of changing in logics of coding of software product.

This type of testing called regression testing which is done after errors or changes fixed which is usually done by the automation test tools. The manual testing is done by the human. The tester put all the test data manually and follow the testing steps mention in the test document for the individual test cases and execute the test. The same way for automation testing the software tester write the automation test scripts in automation tools and run the test. The automation tool executes the test itself and fill the test data and follow the steps according to test document by prepared test scripts. The tester will create a group of test and suits for the tests and save it for future execution. So whenever tester require to execute all the tests again because of error fixing or requirement changes they can simply choose the groups and suits form the saves file and run it. So, automation tool will do it faster and accurate than human manual testing. It will save the cost and time if the deadlines are near with accurate result and generate the detailed test report. [\(Kaner, 1996\)](#).

The test automation required the good amount of money and resources, but test automation will increase the effectiveness and efficiency of software testing and the customer will get quality in service and product. Also, the automation testing requires the staff who have different level of technical experience, deep knowledge about frameworks, proficiency and training. The automation testing tool selection is

depending on the technology the software product under test is built on. The correct automation tool, resources, testing processes, test coverage, documentation and framework are most important factors for the automation to be successful. [\(Kaner, 1996\).](#)

5.2 Automation testing process:

The automation testing processes are divided into five major steps which are as per followed.

- Testing tool selection.
- Define the scope of automation test.
- Planning, Design and development.
- Test Execution
- Maintenance.

a) Testing tool selection

The automation testing tool selection is mainly depending on the technology used to develop the software product. There are so many open-source automation testing tool available on internet like selenium web driver, Squish, Jenkins, docker, Junit, maven, and robot framework. The automation tool should be choosing according to available resources, experienced staff, cost and the framework. It is really good practice to conduct a proof of concept of tool when software product is under test. [\(automation-testing, n.d.\)](#)

b) Define the scope of automation test

The scope of automation test is the mostly depending on technology and design and structure used to develop the software product. The scope of automation is the area, module or part of the software product which will be needed to automate. There are several points which can help to find out the scope of automation testing. Usually, software test engineers like to automate the complex test case which they need to run several times, and which are the time consuming.

Moreover, test engineer like to automate the part which are the main functions and features of the software product in sort the test cases which are marked with high priority and high severity. Also, they would like to add the test cases which are related to large amount of data and large amount of software product users. They can include the test cases which can use to test for cross browser testing. Mostly scope of automation testing engineer chooses the performance of software product like stability, speed and scalability as well as the technical feasibility, network topology, security and code review.

c) Planning, Design and development

This is most important phase for automation testing. This is the same way as software development. Mainly engineers create proper planning, design and development documentation to develop the automation test. In this documentation they mention all the necessary information which will be very important and useful during the whole automation testing process like the testing tool which are already selected for the module of product, In scope and out of scope of automation test, feature to be tested, automation test frameworks, automation test cases, Assigned automation test developer and automation test engineer, schedule and deadlines for the preparation and execution of the automation test, deliverable, resources and remark. So, without proper planning and development documentation automation test is hard to perform. ([automation-testing, n.d.](#))

d) Test Execution

This phase of the automation testing includes the actual run of the developed automation script prepared by the automation test developer. It could be run by the automation developer or tester. The script could include the test data and the action declaration to perform the action by automation tool. Automation tool should be connected with the test management tool where the set of test case are prepared. In automation test tool the test suits contain set of test case and set of test suits are contains by the test group.

The tool will select the group to run the set of test cases as per selected system configuration and perform the test. After run the automation tool will present the report of the automation test evaluation where all the pass are usually reported in green colour and the failed once are reported in red. The test management tool will take result automatically from automation tool and mark pass and fail tests. The test which are already passed that's good result, but which are failed we need to check from the script side that if the automation test developer wrote the correct automation script to perform the action declarations and test data or this is a real bug from the software product side. Usually best to check the test case afterwards manually to perform to check where is the problem. If there is a problem form the script side, they can create an issue in internal work tracking system and assigned to particular automation test developer as well as the same way if it is real bug in software product it will assigned to fix to the software development team. In majority organizations the automation test execution is usually run at night to save time once the execution run the tool will provide us detailed report for all tests. [\(automation-testing, n.d.\)](#)

e) Maintenance

The test automation maintenance phase is carried out after the test execution is successfully done with all planned tests. This maintenance phase usually to done after the test evaluation report checking. The test report presents the detailed information regarding the pass and fail test criteria. Maintenance approach is carried out for the fail to verify the error side. Either it is from the automation tool side or from the software product side. In case error occurs form the tool side that means the written script is not appropriate for that test and it need to change in the scripts. On other side some time we need to add some more script after the changes occurs or new functionality added. It's needed to review and maintained to be sure more about improvement and effectiveness.

6. Automation test framework

6.1 Basics of Automation test framework:

The software product is developing according to the different phases of the software development life cycle. The developers and tester are the usually working on the different module of the same software product. If they use automation framework it will be helpful otherwise every developer and tester will test according to their different approach and methodology and it will cost us more efforts.

Automation framework will be helpful to avoid this kind of situation, and everyone will work on the same platform, methodology and approach. This practice of testing with automation framework will deliver us accuracy in testing, scalability, low amount of maintenance of automation tool, less scripting, reusability of code, stable environment, and low budget of coverage. Also automation framework helpful in maintaining the consistency in test development life cycle and its keep improving the test structure.

Moreover, other benefits of automation framework like non-technical tester can also easily understand the process of automation testing and it will reduce the cost and time for training the staff for automation testing. [\(Desikan, 2006\)](#).

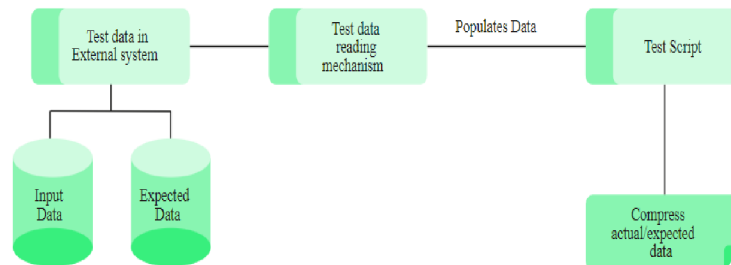
6.2 Types of frameworks used in automation testing:

There are multiple types of frameworks that used in the automation testing. The most common framework that are used in the automation testing processes are divided into different types which are listed below. [\(Desikan, 2006\)](#).

- Data Driven Automation Framework.
- Keyword Driven Automation Framework.
- Hybrid Automation
- Module-based test framework.
- Behaviour driven development.
- Library architecture Test Framework.

a) Data Driven Automation Framework

Figure-6.2.1



[\(6-most-popular-test-automation-frameworks, n.d.\)](#)

The automation test developer usually creates a script with data set as they do in module-based and linear automation test framework. Sometime testers need to test the same functionality with the different data set. The data driven automation framework will save the data set separately in the external database to avoid hard core coding. The script logic will be without the dataset. The dataset will be store in external data base with the key value pairs. So whenever tester needs to run the test script with the dataset they will use and add the key value pairs for the dataset in script logic. [\(Desikan, 2006\)](#).

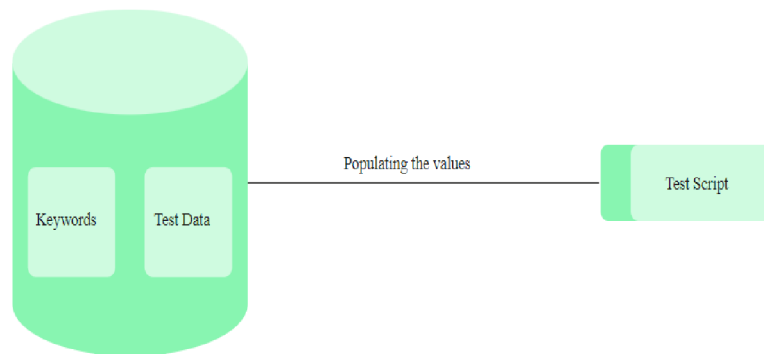
Because of this kind of automation frame the script will be not more complex and reduce amount of coding. It will be efficient for the flexibility and maintenance of automation test script. So, the change in test data set will not affect the scripting. Thus, we can use same scripting for multiple datasets. The data driven framework will allow automation developer to change in input and out parameters in automation test script and use the data from the external source like excel file, docs file, SQL database, csv or any other repository. It will help to execute the test case with multiple data set as the same time to save the time.

The data driven automation framework take a lot of time to set it up properly. Moreover, it required highly skilled person with proficiency in multiple

programming language because they need to know about the external data in advance level and have a knowledge to connect external sources with the framework to execute the test using appropriate data. So according that they need to develop the automation test script for accurate test result. [_\(6-most-popular-test-automation-frameworks, n.d.\)](#)

b) Keyword Driven Automation Framework

Figure-6.2.2



[_\(6-most-popular-test-automation-frameworks, n.d.\)](#)

The keyword driven automation framework is the further step after the data driven automation test framework. As per data driven automation test framework the test data is separately stored in the external database. In keyword driven automation test framework the test data is separated from the test script and store externally with specific code set which is known as the key word which relates to the test script in the external separated file.

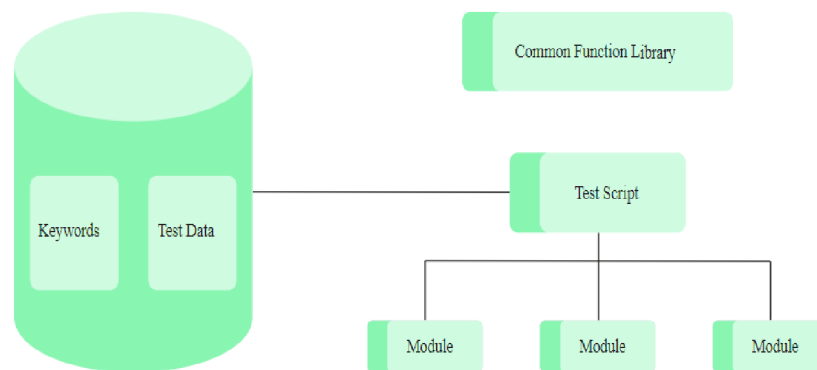
The key word and test data will be store in tabular format. The main advantage of this framework we can use single keyword to execute the multiple scripts. The keyword will be part of the test script because it will contain the various actions to perform the GUI test on software product. So just because of keyword will

not need to do more coding and the script will be simple and less complex and it can be reusable.

The test script developed in keyword driven framework will not be reusable for the other software product it can be suitable only for the particular software product under the test. [\(6-most-popular-test-automation-frameworks, n.d.\)](#)

c) Hybrid Automation Framework

Figure-6.2.3

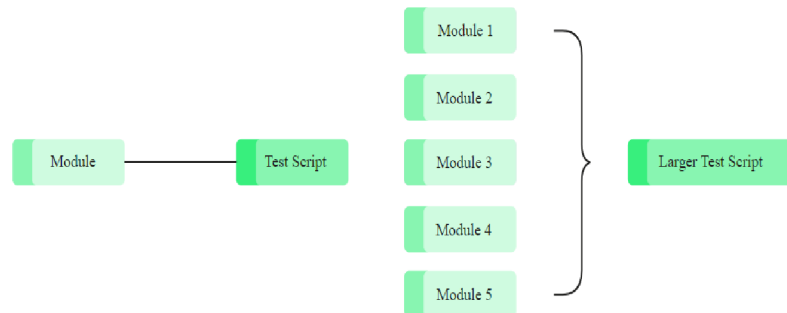


[\(6-most-popular-test-automation-frameworks, n.d.\)](#)

Hybrid automation framework defined as the combination of more than two different test automation framework. The hybrid test automation framework is bit similar as the keyword driven framework. We all know that the world is changing very faster most in IT factor. The testing and developing processes are became integrated with more than software and services just to take the advantages of each other and transfer the weakness so can get the best result of the combined framework. [\(6-most-popular-test-automation-frameworks, n.d.\)](#)

d) Module-based Test Framework

Figure-6.2.4



[\(6-most-popular-test-automation-frameworks, n.d.\)](#)

The module-based automation framework is defined as the software product should be divided into the different isolated module for preparing the automation test script. The automation test script will be separated and individual for all module by the automation test script developer. The developer will develop the script as each module script can be included and join in more extensive test script. The main aim to use this module-based automation test frame because of separated module if there are any changes or code correction needed in the any module. It will not affect the other separated module of software product. So, changes in one module will not generate the unnecessary errors in the other module. The stability and scalability of this framework is advantage for product where the complex level of modularization needed. It will be easy for changes and effective for the maintenance. Moreover, the changes occur in one module will not affect in other module so to find and solve the error will be easy and save the time. Thus, the change in test data will be affect the test script in this framework and it will be requiring the manipulation in the script logic. In this automation framework to creating test case will take less time because we might can use the same test script for another module as well. The module-based automation test framework will need staff who has high skilled programing knowledge. [\(6-most-popular-test-automation-frameworks, n.d.\)](#)

e) **Behaviour Driven Development Framework**

The behaviour driven development framework aim to allow automation testing of functional validations in understandable, readable, and convenient format for the team members who relate to project like tester, developer, business analyst, project managers, stakeholders, scrum masters and other team members who are responsible person in the software development life cycle.

The components of Behaviour Driven Development Framework are as per below.

- **Test data**

The test data is the input data which will be include in scenario to test. It can expect value compared with the actual result of the software product test.

- **Object repository:**

Object repository is defined as the group of locators which are associated with the web element.

- **Build tools and continuous integration:**

The tool available which can create test reports, notifications emails and store authorization credential information.

- **Readers/Generics/Program logics:**

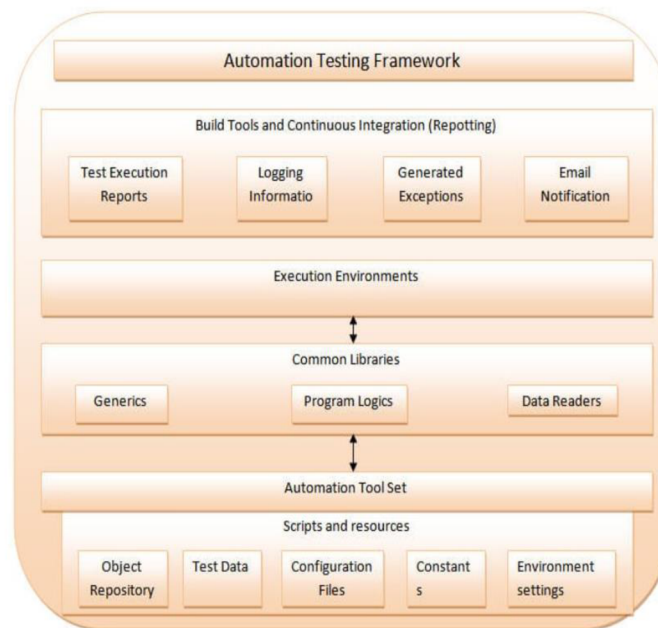
The classes and functions which are created for used commonly in whole framework environment.

- **Environment set up/ Configuration file/ Constants:**

This is information about the version of build, application, specific platform, browser, automation tool and configuration files which will stay static for whole framework. [\(6-most-popular-test-automation-frameworks, n.d.\)](#)

Figure-6.2.5

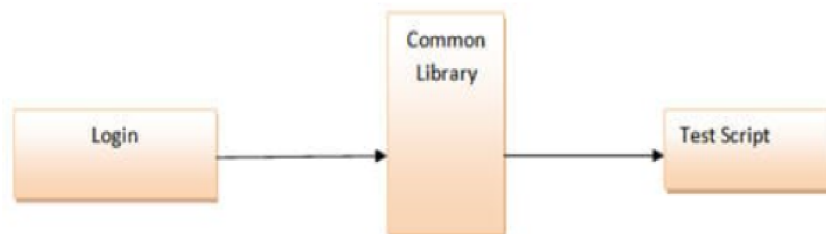
Components of Automation Testing Framework



[\(test-automation-frameworks-selenium-tutorial, n.d.\)](#)

f) Library architecture Test Framework

Figure-6.2.6



[\(test-automation-frameworks-selenium-tutorial, n.d.\)](#)

The library architecture automation testing framework is similar to the module-based automation testing framework. In module-module based framework they divided the software product in different module but instead of dividing connect the software product function by the common functions so that can be used by the other module of the software product.

The library architecture framework is foundationally and methodologically built according to module-based framework but with the additional advantages. In library architecture test framework, it will create a common library from common function of the software product so they create a script according to that library and they can call the library from the script whenever they required to execute the function.

Here they can use the function for multiple time. So basic methodology is to identify the common steps and action and make a group and create a function and relate it into the test script and execute it whenever required. So we do not need to write a code in script multiple time for the same group of actions and functions are reusable. The weakness of the library architecture automation test frame work is it could be bit complicated and need of high level of modularization. [\(Step by step automation in selenium, n.d.\)](#)

6.3 Difference between manual testing and automation testing:

No	Manual Testing	Automation Testing
1	The manual testing is performed by human efforts. Human performed testing by given test step and test data.	The automation testing is performed by the automation tool. Humans prepare the scripts and run it in automation tool, tool perform the testing according to the script developed.
2	Manual testing is not accurate because it human can make mistake, so it is less reliable.	Automation testing executed by tool, so it is accurate and very reliable.
3	It can cost time because of human efforts.	It cost less time because tool executed testing faster than human.
4	Manual testing is good option when frequent repetition of test case execution is not required.	Automation testing is good option when the frequent repetition of test case is required.
5	Manual testing is good for User interface testing because it needs user observation.	Automation testing is not good for interface testing because tool will just follow the script.
6	Manual testing is not reliable for regression testing because code of software product is frequently changes will not provide the accurate test result.	Automation is best option for regression testing because of code changes is frequently changes will provide accurate result.
7	We can apply manual testing on almost every software product.	We cannot apply automation testing on all products. Its need suitable software system or software product.
8	Manual testing can define whether the automation testing is necessary or not.	We cannot fully test the process by automation testing. Automation testing is not 100 percentage possible.
9	Manual testing can perform without any step or data like exploratory testing or ad-hoc testing.	Automation testing cannot perform without specific test step and test data. So testing types like exploratory testing and ad-hoc is not possible.

10	The investment required for the human resources.	The investment required for the tools and automation setup.
11	Manual testing can perform by the non-technical tester also.	Automation testing can perform by highly skilled automation test engineer.
12	Manual testing is not feasible.	Automation testing can do perform performance test, load test, stress test, endurance and scalability etc.
13	Manual testing parallel execution possible with additional human efforts.	Automation testing can be parallel perform on different browser or system or machine.
14	Manual testing can be time consuming and difficult for build verification testing.	Automation testing can easily perform build verification testing.
15	Manual testing can be possible without programing language skill. Tester just need to have proper test documentation to perform testing.	Automation testing can be possible with specific programming language skill. Automation engineer needs to have proper test execution setup to perform testing.
16	The test report is generated by the tester after manual testing done. The test report can be recorded in docs or excel format which is not convenient to all the parties connected with the project like skate holder.	The test report is generated by the automation tool itself. So everybody who are connected with the project can get log in id and passwords and check it any time and it is more convenient and user friendly to check it.
17	The manual test can be possible only with functional testing types.	Automation testing can be possible with both functional and non-functional testing types.
18	Batch testing cannot possible in usual manual testing.	Multiple type of batch testing is possible in automation testing.

7. Performance testing

7.1 Introduction to performance testing:

The performance testing is non-functional type of testing. The performance testing verifies the performance of the software product or application in different expected workload. This testing type's main aims to determine the performance of software product's speed, stability, scalability, reliability and response time. [\(Kaner, 1996\)](#).

The performance testing mainly testing about speed, stability, and scalability.

- **Speed**

It is verifying for the response time of the application or software product. It means the software products working speed on the given actions.

- **Stability**

It is verifying behaviour of software application in different workload. This testing done with the under the different workload and check the application is staying stable, or it is broken because of change of workload.

- **Scalability**

It is verifying the behaviour of the software product with the number of User using it at the same time. It is checking the maximum number of User's workload can handle by the software product.

The matrices and measurements required to better understanding about the quality and effectiveness of the performance testing. To make the changes without matrices and measurement is quite thought and costly in case of any potential error occurs in middle of performance testing process.

Measurements can be defined as the data collection for the process which is used to measure while in testing process like respond time.

Metrics can be defined as the calculation which can use the measurement and present the quality of result, e.g., Average response time.

The testing steps for performance testing are different and depend on organization as well as the software product or application under the test but the main goal of performance testing is almost the same. [\(Kaner, 1996\)](#).

7.2 Measurements used for performance metrics:

Performance testing processes have several ways to measure speed, stability, and scalability for performance metrics. As well as there are few more measurements which are used by the performance metrics which are as per below. [\(Types-of-performance-testing, n.d.\)](#)

- **Memory usage**
Memory capacity available on the processor to run the software product.
- **Processor usage**
The processor takes the time to process certain tasks to run the software product.
- **Response time**
The amount of time to send request and respond to it by application.
- **Request**
How many requests can be handled by software product within a second.
- **CPU Utilization**
The time taken by the CPU to process the request.
- **Throughput**
The amount of bandwidth used in test process by the software product. It is measured in KBPS.

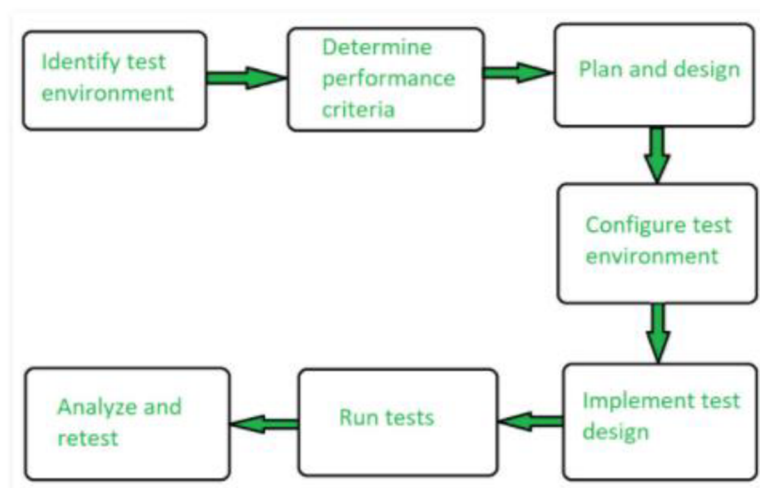
- **Thread count**
The software product health can calculate by the number of threads running and in active status.
- **Garbage collection**
The memory supposed to return unused memory back to the system memory. The garbage collection supposed to monitorized to unused memory will return to system back.
- **Committed memory**
The virtual memory usage.
- **Bandwidth**
The bit used per second by the network interface.
- **Private bytes**
The bytes allocated by the process which is not used or allocated by the other processes.
- **Disk time**
The time used by disk to read and write the request and process it.
- **Maximum active session**
The maximum session can be active at the same time.
- **Top waits**
Identifying what wait time can be cut down as per the memory can be fast retrieved from memory.

7.3 Performance testing processes:

There are multiple types of processes that used in the performance testing. The most common processes that are used in the performance testing are divided into different types which are listed below. [\(Marick, 1995\)](#).

- Identify the test environment
- Determine performance criteria.
- Plan and design.
- Configure test environment.
- Implement test design.
- Run tests.
- Analyze and retest.

Figure-7.3.1



[\(Performance-testing-software-testing, n.d.\)](#)

a) Identify the test environment

This is the most important step for the performance testing is to identify the test environment. Before start performance test its need to check availability of production environment, test environment, skilled performance test staff and the automation testing tool. Also, the proper documentation for the scope of the performance test is required.

Moreover, the better understanding of software, hardware and the network configuration is playing the vital role to get the more efficient test result by the performance tester. [\(testing, n.d.\)](#)

b) Determine performance criteria

To determine the performance criteria is important to have knowledge about the scope of performance testing. First of all, the tester should identify mainly from the project specification documentation about the constraints, aim and the obstacles. As well as the skilled performance tester should empower his skill according to available environment and testing tools and need to set the wide range of test criteria and benchmarks. [\(Kaner, 1996\)](#).

c) Plan and design

The planning and design include the schedule of performance testing as well as the test data and test cases matrix. The detailed information about when and who will do the performance test also the scope of the performance testing. It is determining the planning and design for test of all possible use cases with the available test environment and tool.

d) Configure test environment

In this stage tester need to verify that the test environment is properly set up for the test and already system configured with the testing tool. Also, the other necessary provided resources should be up to date.

e) Implement test design

The implementation of test design plays very important role. The performance tester implement test according to test design. Test design development mostly depends on the test plan and test strategy. The implementation of the performance test has been done with the resource available and the scope of performance testing. [\(Kaner, 1996\)](#).

f) Run tests

In this phase run the prepared test cases and evaluate the result. After evaluating result is should be stored in proper test execution document. This documentation could help in future projects for help and recommendations. [_\(testing, n.d.\)](#)

g) Analyze and retest

This phase is same as the maintenance phase. While running test, tester needs to analyze is test running as well as the result. In some testing tool it is good practice to set inspections so it will take the screenshot of all the necessary stages after each other. Also, in case of test failed tester needs to analyze the fault. Either it is cause from automation tool or form software product which is under test. Tester will analyze the problem and fix it and it will prepare the failed test case for the retest. [_\(Kaner, 1996\).](#)

7.4 Good practices and recommendation for the performance test:

There are several helpful recommendation as well as some good practices that could help user in the performance testing. The most common and effective recommendations are listed below. [_\(Marick, 1995\).](#)

- 1) Most important to identify the tool which is best suitable for your performance testing as well as the software product which is about to go under the testing.
- 2) Tester should keep the performance testing environment separate from the User acceptance testing process environment.
- 3) Tester should change the environment before or after test run and analysis done. If the tester would change the test environment between the test run it could mess with the test evaluation report. And the rest report will not be that accurate.
- 4) The tester should remark the reason of failure test in document.

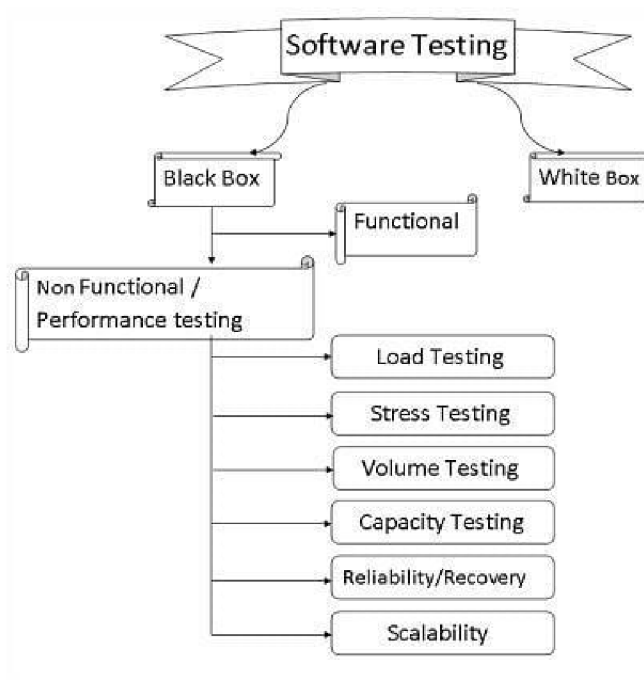
- 5) The tester should change in test as per the change in application or software product build change or the test environment is updated or changed in system configuration.
- 6) The tester should compare the performance testing results for the same software product or application on the different platform, system or operating system.
- 7) It is a good recommendation for the testers that to make the test summary report after each test run so comparison of test result will not be compared all the time when the test result will be referred.
- 8) The tester should include the following points in the performance testing documentation.
 - Detailed description of the performance test
 - Purpose of the test
 - Scope of the testing.
 - Response time
 - Duration of test.
 - Remarks on failed test.
 - Details about system configuration.
 - Details about the automation testing tool.
 - Stability report.
 - Scalability report.
 - Metrics, Graphs or Graphs comparison.
 - Recommendation.

7.5 Types of Performance testing:

There are multiple types that are mostly discussed in the performance testing. The most common types used in the performance testing are divided into different parts which are listed below. [\(Dyer, 1992\).](#)

- Load Testing
- Capacity Testing
- Scalability testing
- Volume Testing
- Stress Testing
- Reliability testing

Figure-7.5.1



[\(Types-of-performance-testing, n.d.\)](#)

a) Load testing:

The load testing is the type of performance testing where the application is going under the different amount of user load. This testing type verify that the application or software product's behaviour under the high and low amount of user load. Moreover, it's how the software product or application can respond

upon the user's request as well as how simultaneously it can handle the multiple users request at same time.

The main goal of Load testing is to find out the about stability and simultaneous functionality of software product before it delivers to the customer. In load testing tester verify the software product respond time for each action at different time as well as how the software components functioning which are connected to each other with the various load. It is also checking the performance of the data base component because the load testing mostly done for the client-server and web-based application. As it is reliable testing type for client-server it is also verifying the network connectivity if it is delay or stays simultaneous.

Load testing result is also depending on the system configuration on which we are running the software product like the control processing unit capacity, memory size and network configuration. Load testing report evaluation can help to reduce the cost of failure and project team can be ready with the maintenance criteria. So the company can deliver the better software product and win the customer heart and increase in his satisfaction. [\(Types-of-performance-testing, n.d.\)](#)

b) Capacity testing:

Capacity testing is type of performance testing where testers verify that developed software product can handle the amount of user and the transaction that meet the business requirement as it is specified, As well as the software product environment is also capable enough to handle the user traffic and transactions as it was designed for.

Capacity testing is playing a vital role to avoid the potential problems which could appear in future. So, tester can have a better visibility about the software products limitations, and it could help to plan the other performance testing like Stress and Volume testing. Online banking and Amazon online shopping sites are the best examples for the apply capacity testing and it could play a major role to know about software product performance. Change software product environment like disk capacity, memory usage.

c) Scalability testing:

The scalability testing is a type of performance testing which is ensuring the performance of software product when number of the users are scale increasing and decreasing. The scalability testing is performed in the particular test environment where they can test the software product with the large system and huge amount of load.

Scalability testing ensuring the scale point where it can handle maximum amount of load and after identifying that which point the software product will stop responding. There are some key factors of the software product which usually measured by the scalability testing which are number of users, transaction of data, data volume, network usage, CPU usage, Memory usage, responding time, task execution time and many more. The scalability testing strategy can be different as per the software product is being tested.

d) Volume testing:

The volume testing is type of the performance testing where testers identify the behaviour of software product performance while volume of data varying simultaneously. In this test the large amount of data is uploaded on the database which relates to the software product and gradually increasing in volume of data.

Testers are verifying the behaviour of the software product while the process and check it if the exchange of the data can any impact on the performance of the software product. [\(Types-of-performance-testing, n.d.\)](#)

e) Stress testing:

The stress testing is a type of performance testing where tester find out the maximum stress level can handle by the software product or an application. The software product is going under the maximum stress, or we can say the application runs under the maximum users load until it breaks down itself. Basically, we can say that stress testing to verify that on which level of stress and how it can break the software product.

Because of this testing tester can identify that amount or range of the load can hold or sustain by the software product or application. This is most important question while executing this test is how to run this test.

Usually, the tester applies maximum load which the software product is already tested, and they are starting the increasing the load gradually and when the server starts not responding, the testers consider this point or level as breaking point of the software product. There is multiple ways to break the system like uploading or downloading large size of files as the same time or more than maximum amount of user are using the same component or unit of the software product under the test. [\(Marick, 1995\)](#).

f) Reliability testing:

Reliability testing is the type of performance testing which is mostly performed in the particular test environment. The reliability testing is ensuring that the software product is failure free during the specific time of period in the defined testing environment. Moreover, we can say that reliability testing is performed to check if the software product is reliable enough and not failure to do specified purpose which are software product developed for in given specific period of time and defined environment.

This testing main aim to provide the good quality and standard product. Sometimes it is ensuring by the executing the same test case again and again to make sure that all the time testers getting the same as expected result and it stays continue all time. This type of behaving is also proving that the software product is reliable.

Reliability testing process helps to measure performance of various component of software product after the change in requirements or bug fixing as well as its uncover the main root of errors. The reliability testing is divided in the three parts which are modelling, improvements and measurement. [\(Types-of-performance-testing, n.d.\)](#)

8. Software testing levels

8.1 Introduction to software testing levels:

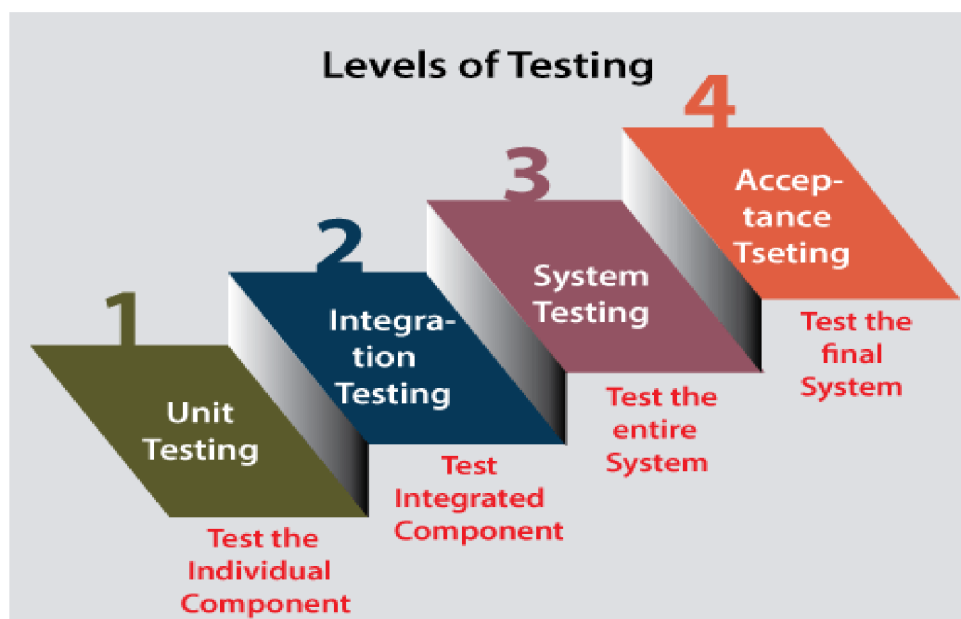
The last phase of Software development life cycles is deployment of software to the customer but before we release the software it needs to be undergoes from some testing levels. All the levels should be performed in the defined orders.

Each testing levels have their own different purpose and importance for software development process. Also, some testing levels have sublevel too. But there are major testing levels are only four testing levels which are unit testing, integration testing, system integration testing and acceptance testing. There are other levels which can be in testing process add according to the testing objectives and prospective of customer. (Vaniya, 2018).

The main four levels of the testing are listed below.

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

Figure-8.1.1



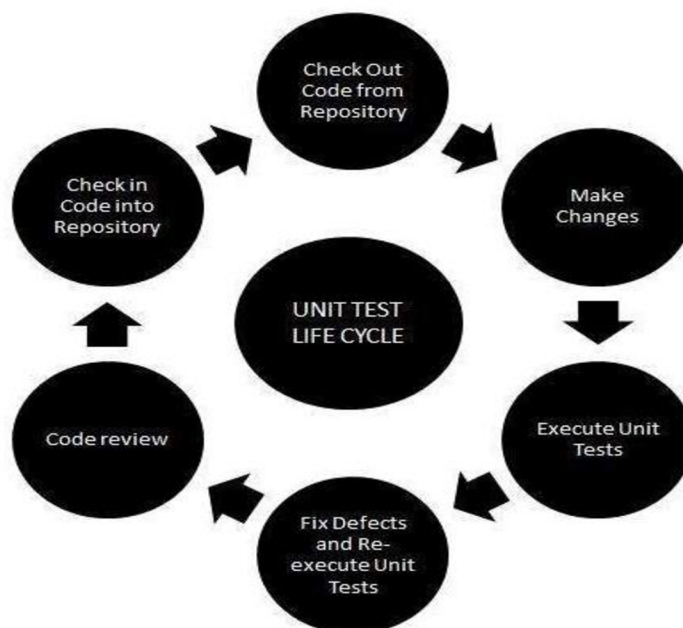
(Google Images, n.d.)

8.2 Unit testing:

The unit testing is first level of testing levels, and it is initial stage of whole testing process. This type of testing performed to verify the functionality of units or module or group of codes and needs to make sure that each part of unit is perfectly functioning. Unit testing test the basic functionality. Unit testing cannot be able to test whole functionality of software but rather it does the component or modules which are working individually from each other. [\(Vaniya, 2018\)](#).

Unit testing help to test internal design, internal logic, internal path, and error handling. Unit testing level includes only a type of testing is unit testing. Unit testing is performed by the software developer who develop the code to ensure that the function is working as per specific requirements as per mentioned in the test document. In early stage with the help of unit testing it could save time and money as well as find out the errors in early stage of development process. Unit testing also help to fix and prevent instead of potential errors which could occurs in future. The main advantage of the unit testing is we can easily fix the small group of code in early stages rather than whole complex code design after development process done. [\(Kaner, 1996\)](#).

Figure-8.2.1



[\(software_testing_dictionary_unit_testing, n.d.\)](#)

According to unit test life cycle it is performing in certain defined steps like executing unit testing first. Afterwards identifying the defect and developer need to fix it and re-execute the unit test. Then next step it to review the code again and after the performing it start processing for upload the code from local repository to main central code repository. Now the code of unit will be running with all the other unit code and performing deferent testing processes on it. In case of failure, it will again require changes in it and again execute the unit on it.

Unit test can be done with different type of testing techniques like black box testing, white box testing and gray box testing. This testing type can be done by manual are automation testing. Manual testing can be done by the manual tester or developer as well as non-technical staff can do it by following the test document. Automation testing can be performed only by the skill technical staff only and with require automation resources. [\(Dyer, 1992\)](#).

8.2.1 Black-box testing:

The tester has not provided or should not have knowledge about mechanism of internal code structure. In other words, tester just test the software product from the front-end side like the functional testing, GUI testing and data driven testing. Black box testing can be done by the developer, tester or any non-technical person. To perform this testing testers, need to have proper test documentation created with the test data and test environment as well as with the detailed explanation of functional test specification. This testing techniques is cost less time among other test techniques.

8.2.2 White-box testing:

The tester has already provided or should have knowledge of the internal code structure. In this testing technique tester required not only the front-end testing but tester also required to do some back-end testing as well. To do some back-end testing tester need to know about the internal struct of code design of software product. White box testing ensures the structure, code and performance of the software product. Moreover, we can say that system

integration testing, database testing, server testing and code-based testing are more suitable examples of white box testing. The white box testing is more complex and time-consuming testing technique than Black box testing. It can be only performed by the experienced skilled tester or automation test developer. White box testing needs well explained test document with test environment setup and skilled testing engineers.

8.2.3 Gray-box testing:

The tester has already provided partial information or knowledge about the mechanism of code structure. In other words, we can say that tester has provided limited information about coding. In this type of testing technique tester testing front-end functionality as well as some part of back-end functionality. To perform this type of testing tester required to have some additional testing skills with programming knowledge. The coding bases and GUI based test cases are best examples of the gray box testing. It is less time consuming than the white box testing technique as its test case can be done in short period of time. [\(Kaner, 1996\)](#).

8.3 Integration testing:

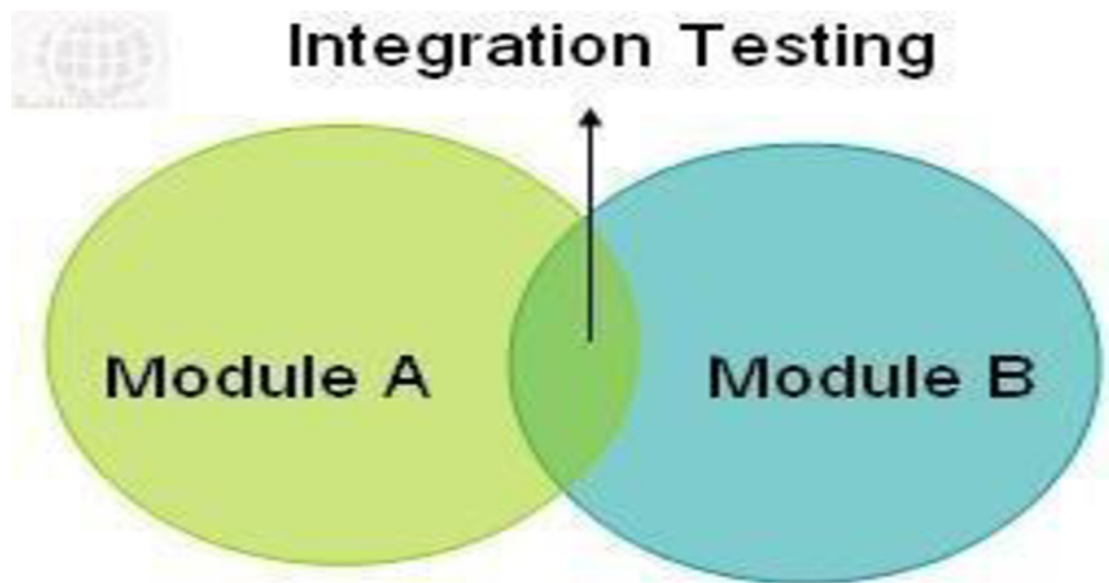
Integration testing is next step after unit testing. Integration testing performed to ensure the functionality between component and software design. The integration testing main aim to verify the interface between more than one unit or module with software design or system. The software components, unit, system, or databases are integrated between each other by iterative way or all together communicate. [\(Vaniya, 2018\)](#).

This testing required to perform after unit test because testers need to ensure first that how components work individually and after in integration testing testers verifies how it works together. The integration testing makes sure that the software is working in a well-organized and competent way. The verification of the component integrity should be carried out immediately after unit test so finding and fixing the bugs at early

stage will be became easy and help to make changes in code rather than changes in complex code structure.

A best practice to perform integration testing time to time will be very less costly, convenient, time saving and can give us accurate result for system integration testing preparation. [\(Dyer, 1992\).](#)

Figure-8.3.1



[\(Integration-testing, n.d.\)](#)

The integration testing becomes very important some time because some developer fix the unit test error but do not test it again and without testing, they deploy the code to main branch. In case of that error is not really fixed by developer it could be detain in the integration testing. Also, one of the most important reasons to perform integration test is that we all know in development phase the whole software is separated in small part and divided development work to the different responsible developer.

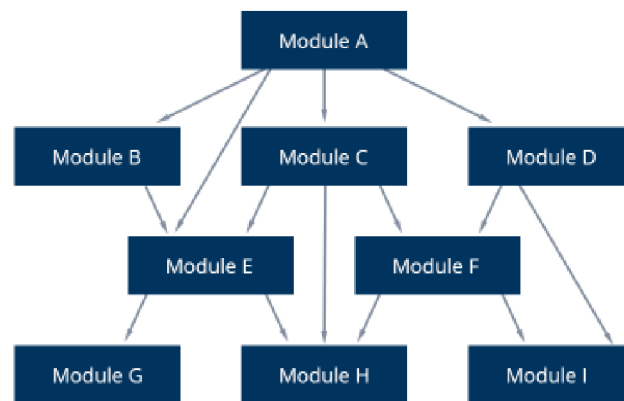
All the developer are starts working on the given particular part of the software product. Meanwhile they are working on the same platform for coding, tool, programming language, databases and operating system but while developing they applied their coding logics, designs and mechanism. Here comes the main requirement of integration testing after all the separated software product part

integrated with each other and the tester verifies that they are all working as per requirement using the system interface and detects the bugs in case of code mechanism not communicate well because of interface.

There are few types of integration testing approaches as per below. [\(Integration-testing, n.d.\)](#)

8.3.1 Big-bang approach:

Figure-8.3.1.1



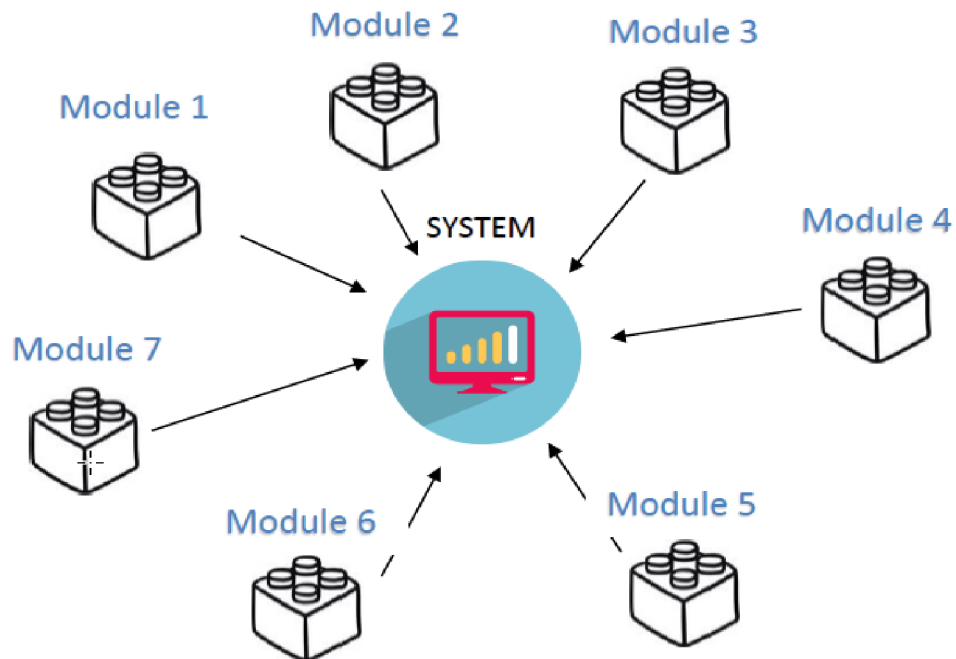
[\(Google Images, n.d.\)](#)

The big bang approach includes the whole set or group of integrated modules of components together at the same time. It is not testing the module integrated one by one. It will be ensuring that everything which is integrated together and being tested at the same time can be find the root of problem. But it is difficult to find out that in which module the error has occurs.

The main advantage of the big bang approach is that it is most suitable for the small software product instead of large. This approach can be becoming very time taking process for the large and complex problem because as per the big bang approach principle it will integrate whole system and run together so at this moment it hard to find out the bug and fix it in a way where is will not hard other part of the software product. So, in this case this approach can be proven costly. [\(Software_testing_levels, n.d.\)](#)

8.3.2 Incremental testing:

Figure-8.3.2.1

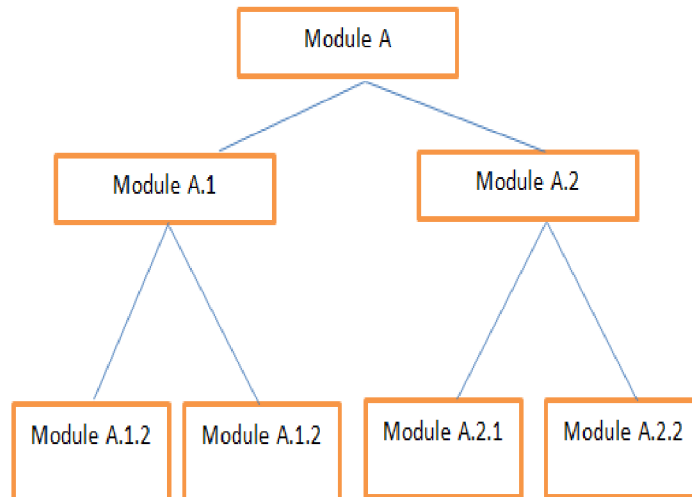


[\(Integration-testing, n.d.\)](#)

The incremental testing approach works like it integrate the more than two module which are logically and according to design of development connected and after it is tested for the actual functionality of software product. As well as in same manner new modules are incrementally integrated one by one in continues process until all the logically related module are integrated with each other and tested for fully actual functionality of software product. The developer creates some dummy program to facilitate the software integration testing activity and they are known as stubs and drivers. Stub is defined as the module which is under the test and the Drivers are defines as the module which to be tested. Stubs design use in the Top-down approach and Diver design for bottom-up approach. [\(Software_testing_levels, n.d.\)](#)

8.3.3 Top-down approach:

Figure-8.3.3.1

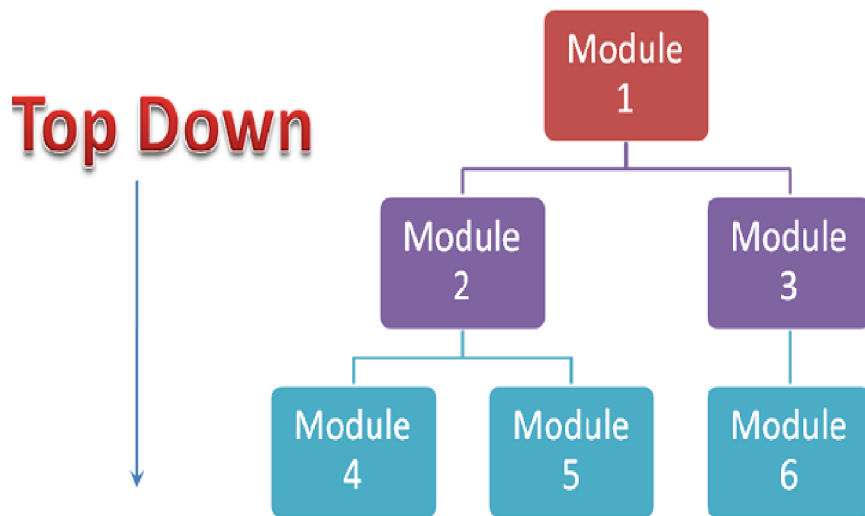


[\(Google Images, n.d.\)](#)

The Top-down approach test top-most and important components first and it gradually move down and test the components as per the priority level one after one. As per the figure below we can see that the module A is firstly tested independently and then module A.1 and A.2 integrated with module A and perform the integration testing for the actually functionality and logically connected modules are working well or not.

After the successfully integration done between the module A, A.1, A.2 we can clearly see that module A.1 is connected logically for further testing process with module A.1.1 and A.1.2 and in the same manner in module A.2 is following Top-down approach and gradually move down and connect the new module A.2.1 and A.2.2. and this module are now ready for the further integration testing, and this is the way how top-down approach is performed for integration testing. [\(Software_testing_levels, n.d.\)](#)

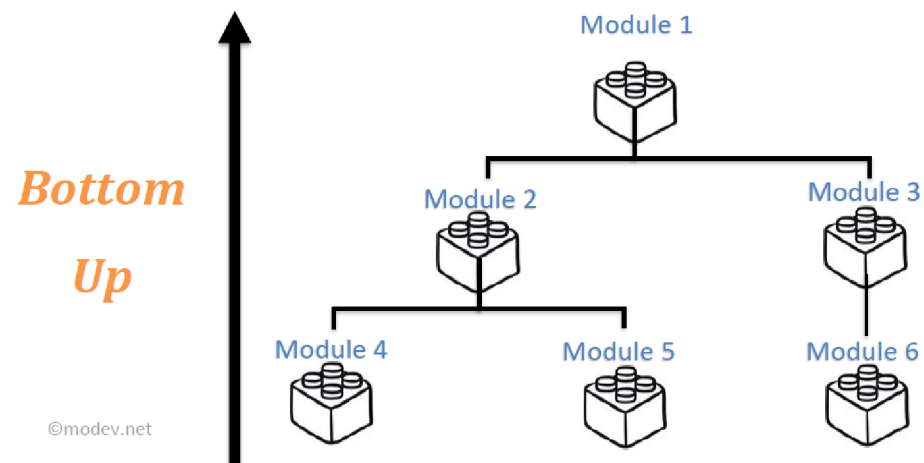
Figure-8.3.3.2



_(Google Images, n.d.)

8.3.4 Bottom-up approach:

Figure-8.3.4.1



_(Integration-testing, n.d.)

The bottom-up approach is similar approach and opposite of top-down approach. In bottom-up approach test first the lowest unit and it moved up gradually one by one. This approach gives important to lowest modules and less priorities module of the software product.

The main advantages this bottom-up approach is that in case of errors occurs in the lowest module it can be detect and corrective action can be taken on priorities level. By performing this approach, it is easy to find out the fault better than big bang approach like we need to wait for all modules to develop properly to perform the big bang approach.

According to figure mentioned below, we can clearly see that the module 4, module 5 and module 6 are the lowest module which are already unit tested separately as the topmost modules are usually done unit tested separated priorities levels in top-down approach. In picture we can see that module 2 and 3 are the gradually connected in going to up forward. Suppose the module 2 and module 3 is not yet developed for integrate with the module 4, 5 and 6. So the logical functionality of the module 2 and module 3 cannot logically call to module 4, 5 and 6. So for that reason developer develop the program which can “Drivers” to simulated with the module 4, 5, and 6 and it will call them.

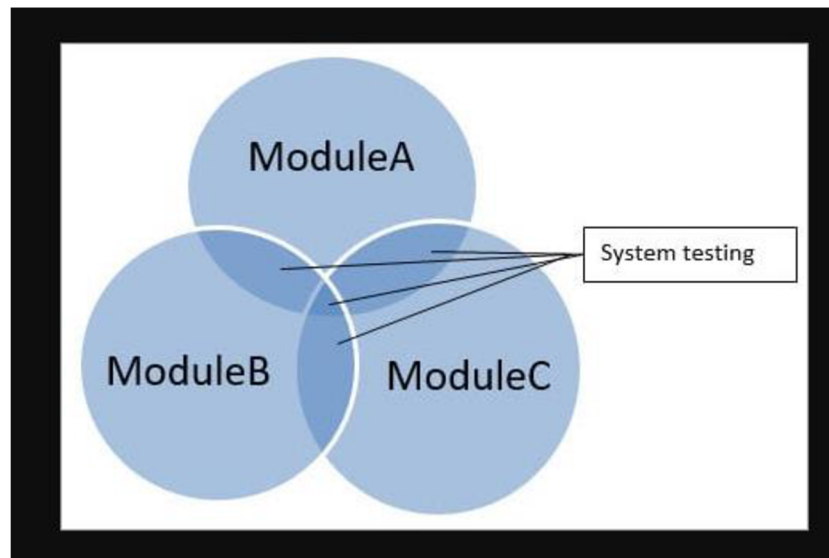
In other words, we can define the “Drivers” as the dummy program which will use to call lowest function when the calling functions are not developed. This “Drivers” are the most important input to create an interface for the module under test in bottom-up approach integration testing.

8.4 System testing:

System testing needs to perfume to ensure that the whole system is working as per the business requirement specified in the test documents. The system testing can be including the log in process, creating, sending request, getting request responses, search, edit, delete, add, send, upload, download, selecting and ending actions. It required to appropriate use of software product which all parts are completely developed and ready to test for the system testing.

In other words, the system testing defines as the testing process which involved the complete and fully integrated software system which will evaluate and verifies the end-to-end system specifications. [\(Vaniya, 2018\)](#).

Figure-8.4.1



_(Google Images, n.d.)

System testing could be performed by both ways like test automation testing and manual testing. This testing is verifying the whole software at same time. To perform system testing tester required specific type of test environment to test the software product. System testing includes set of testing type which can help to evaluate the software from all testing criteria.

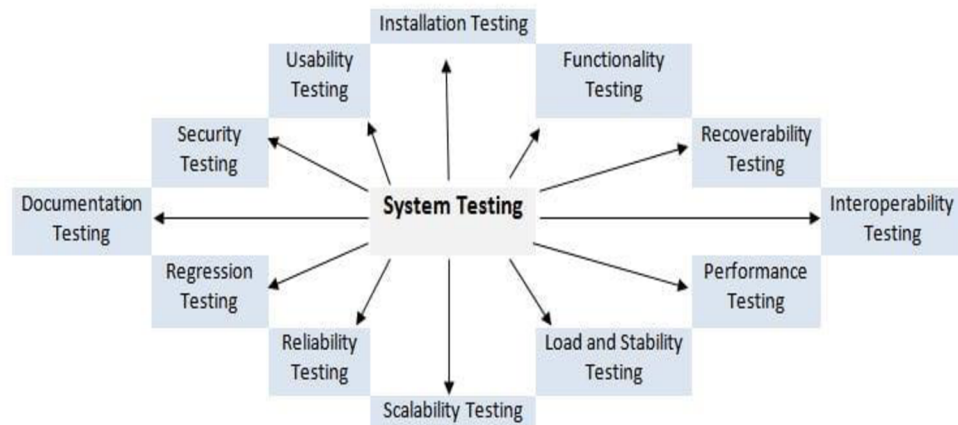
We can say this is the last level of testing because after this testing process done the customer will test and verify the software product before they took handover of it. System testing is most important type of testing among all testing types because it includes all functional testing, non-functional testing, interface testing, network testing, performance testing and security testing. [_\(testing S. , n.d.\)](#)

The system testing evaluates by the end-to-end scenario to test whole integrated system at same time so very high changes to cover the unwanted errors during this testing. This testing usually performed in the same environment as the production environment so they can test it according to the user prospective and take some necessary actions against unwanted situation before it delivered to use in public domain. This testing also includes the testing criteria form the system requirement like hardware requirement, system installation, system configuration, network domain as per mentioned requirement in the documentations as this testing could cover the design and

architecture side and business requirement side. Let's have a look on the testing types set which is include by the software system test._(Dyer, 1992).

8.4.1 Testing types:

Figure-8.4.1.1



_(System-testing, n.d.)

1) Functional testing:

Function testing is defined as type of testing which performs to ensure the functionality against the software system. It is validating the functional requirement of software product as per the specification defined in business requirement. This testing takes performs with given input data to get output and verifying it with the expected results. Functional testing mostly performs black-box type of testing only. This testing can be performed either as manual testing or automation testing.

2) Non-functional testing:

Non-function testing mostly about to verification of software product's speed, stability, and scalability. Moreover, this testing performs to evaluate the performance, compatibility, reliability, and other non-functional criteria which left to test by the functional testing.

Non-functional testing is tough with the manual approach because it is hard to get the accurate result as well as it could be time consuming. It most efficient and mostly done by the automation tool only. [_\(Types-of-software-testing, n.d.\)](#)

3) Unit testing:

The unit testing performs to test of code of unit or module or component of the software product. It is verifying the developed code of unit or module, or component is working as per the expectation. The unit testing is done right after the development of unit and usually it is done by the developer only. Unit testing is mostly done at the initial stage, and it is white-box testing technique as to perform it required the code knowledge.

4) Integration testing:

Integration testing is type of software testing which verifies the communication between two more individual unit or component of software product. Usually, the whole software development work is divided in small part and assigned to developer to work on it. Every developer work according to his coding knowledge and logic and develop the functionality of the unit or component.

When this all components are integrated to each other and then after Integration testing performs to check the functionality and ensure that they are logically well connected and working as per the specification business requirement together at same time. [_\(Desikan, 2006\).](#)

5) System testing:

System testing define as the end-to-end testing which will test the whole software product where all the modules are integrated with each other and it is ensuring that whole functionality is well integrated and working as per the business requirement. Moreover, System testing is set of multiple testing which are executed in proper sequence to cover all the testing

criteria as well as it is verifying that software product is interfaced well with the other related hardware or software system.

6) Acceptance testing:

Acceptance testing is the last phase of testing process. Usually, it is done by the end user or customer to verify the software product as per the business requirement before the deployment. This testing is done after unit testing, integration testing and system testing. After acceptance testing done the software product moved to the production environment for real purpose use. [_\(Desikan, 2006\)_](#).

7) Smoke testing:

Smoke testing is the one of the initial stage testing processes which is ensured that if software product is ready and comfortable enough to apply further testing processes. Some testing word is came from the hardware testing side. Actually, It was testing in case when hardware are connected with each other and it not catching fire while working and just because of that its knows as the smoke testing. In software testing process some initial test cases are creates for perform the smoke testing to ensure that the software products functionality is stable enough for the further testing processes or not. [_\(Types-of-software-testing, n.d.\)](#)

8) Ad-hoc testing:

Ad-hoc testing is type of testing which is done by random testing without any planning, documentation, or any planned activity. Ad-hoc testing does not follow any structural way or input data or any testing steps. It is fully random testing. Ad-hoc testing is usually performed to break the testing process and try to find out the possible find out error and defect at the early stage of development process.

9) Exploratory testing:

Exploratory testing is all to apply personal freedom, thinking and type to test software product. This testing is done by the tester but without any test

case creation. Testers apply to their individual testing skills to investigating the functionality of software product.

10) Stress testing:

Stress testing ensure the software product stability and reliability. The main aim of this testing is to verify the error handling capacity when the extreme heavy load of user activity applies at the same time. It is found out the breaking or crashing point of the software product in particular amount of load. Stress testing is also performed to find out the behaviour of the application under the abnormal conditions.

11) Sanity testing:

Sanity testing is type of software testing which is performed after the certain case like change in functionality, change in requirements, code changes or bug fixes. Sanity testing is performed on the new build version after this type of fixes happens during the development process. Sanity testing aim that the new bug version is perfectly ready and cover all the necessary things. In case sanity testing fails new build, version will be rejected.

12) Volume testing:

Volume testing verifies the software product behaviour and system responds at the expected certain volume of data. Volume testing usually performs to verifying the software product performance when the volume of data increased in the database. Volume testing help to study the impact on the response time and other performance criteria when certain volume data is applied.

13) Compatibility testing:

Compatibility testing is type of software testing which ensure the software product is compatible to run with on different platform, operating systems, different hardware configurations, mobile operating system, network environment and web browsers. It is nothing but just verifying if

developed software product is comfortably operating as per the business criteria with all this. [_\(Desikan, 2006\).](#)

14) GUI testing:

GUI testing is known as graphical user interface testing. This testing is checking front end side of the website or application. It is ensuring the elements on the screen like if they are working as per requirements or not like search bar, filter bar, radio button, check box, list box, drop down buttons, text box as well as the size and position of these web elements.

GUI testing also verifies the error message, Pop up message, toggle, arrows, icons, text fonts, spellings, colour, size of fonts, warning message, images and check if everything is attractive, well placed and watchable.

15) Security testing:

Security testing verifies the weakness and loopholes in the software product. It is ensuring that if it is possible to break and harm the software product from the outside factors and take some appropriate action to prevent software from outside attacks.

It is also testing about security issues like risk, threats, and hacking kind of the activity. It is also including the security scanning of product as well as the ethical testing activity and penetration testing.

16) Documentation testing:

Documentation testing takes place just after the test documents are prepared for testing activity. The documentation testing verifies the test documents like Test Plan, Test case, Traceability metrics, test scripts and requirements. This type of testing ensures that if these documents cover all testing approaches. [_\(Types-of-software-testing, n.d.\)](#)

17) Usability testing:

Usability testing is type of software testing which ensure the software product is user-friendly or not. It is performed by the end user or any random person to check and flexibility, the user interface, handle controls and if it is easy or hard to understand the graphics of software application.

18) Reliability testing:

The reliability testing is type of software testing which ensure that the software product is bug free and completing it execution of certain operation in specific time period and in the configuration environment.

19) Scalability testing:

Scalability testing is performed to ensure the system ability and behaviour and how it behave in varying the user traffic. It is checking the performance of system or stability of network when the number of users, Request sending, request acceptance, data upload and download like this kind of transaction happens. [\(Desikan, 2006\)](#).

20) Installation testing:

The installation testing is performed to verifying that the software installation file is able to install and uninstall the software product perfectly also it is making sure that it is containing all the configuration files. It is also check that partially, fully, or upgrade version install and uninstall processes.

21) Load testing:

Load testing is type of testing technique which improve the performance bottleneck and working functionality before it go to production development. This testing is performed under the specific expected load and checking the application behaviour under multiple users using it simultaneously.

22) Stability testing:

Stability testing is non-functionality software testing which can be performed to identify the stability of software product in certain cases. This can be checking if software product goes slow down and encounter some functionality related problems. It is verifying the efficiency of the software to work continuously function simultaneously for long time.

23) Performance testing:

Performance testing is most important type of non-functional testing. It is mainly verifying the stability, scalability, and speed of the software product. As well as the response time, reliability, and speed of the software product.

It is hard to done by the manually and time consuming as well as not getting accurate result for it. It mostly done by the automation tools. It is done by the skilled performance tester. Usually they used Jmeter, Loadninja and HP Loadrunner as automation tester. [\(Desikan, 2006\)](#).

24) Recoverability testing:

Recoverability testing is the software testing type which identify the software ability to recover from the crashes or network failures. The main aim of recoverability testing to determine the weather software product can be working continuously after loss of data.

25) Negative testing:

Negative testing is performed to check with false test case which containing the wrong or negative inputs data. The negative testing is preventing the software testing from the negative inputs to crash down and improve its functionality, quality as well as the stability. [_\(Types-of-software-testing, n.d.\)](#)

26) Positive testing:

Positive testing is performed to check with the valid and true test case which containing the valid test input data to execute the test. Positive

testing main purpose is to verify if the product behaves as it is expected to behave and identify that the software product is developed as per specific business requirement. It should give us positive result as per the customer requirement.

27) Globalization testing:

The globalization testing is type of software testing which ensure that the version of the software product is working fine for all the assigned locations. It is checking if it's perfectly functioning with all international inputs, data, culture and the can change the language according to the locations. In sort this testing ensure that the software product can be used worldwide. It is also known as the internationalization testing.

28) Penetration testing:

Penetration testing is type of security testing which ensuring and testing the loophole and security criteria and find out the way to prevent the software product from the threat and risk of cyber-attacks.

Penetration testing can be done both way manual testing and automation testing. Penetration testing is known as the Pen test also. Pen test can be done by the professional skilled penetration tester and it expensive in budgets

29) Regression testing:

Regression testing is type of software testing which is performed after the changes in code or fixes in errors has been done. This type of testing executes the set of test cases which are already executed before.

The regression testing ensures that other functionality of the software product is not affected due to change in code somewhere in the software product component or because of bug fixing. As well as the regression testing takes placed also after added some in feature to the software

product to detect and fix the performance related issues. Regression testing can be done by the manual way as well as automation.

30) API testing:

API testing is known as application programming interface testing. This testing verifying the application programming interface if it is communicating well or not. It is testing the functionality, performance, and security of the software product. The API testing can be done by the automation tool and Postman API tool is mostly used to perform this testing. [_\(Types-of-software-testing, n.d.\)](#)

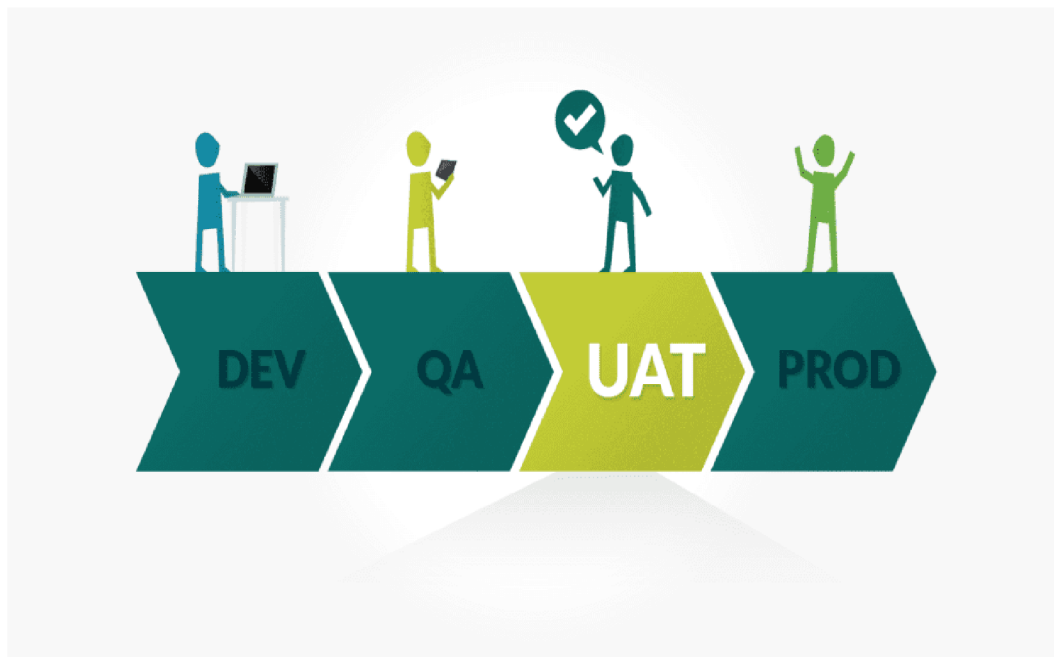
8.5 Acceptance testing:

Acceptance testing is the final stage of testing level of software development lifecycle. It is also known as user acceptance testing. The acceptance testing is performed to ensure that the software of application is ready to release or not to the production environment. There are always many improvements in requirements and functionalities during the development process. [_\(Vaniya, 2018\).](#)

It is very important to verify that is it meet the customer requirement and satisfy the business need before it releases for people to use it. Also, it is also ensured about the functional and non-functional requirement. If there is any problem or no satisfaction form customer, it will not go for release and it stay in development process. It will go back to developers for changes which are asked from the customer. They make changes in development as per requirements and again they follow all stage of testing and come back to the here again for acceptance testing.

Usually, the acceptance testing is performed by the end user or the customer. They are following all the end-to-end process and check all the functionality and business flow work the same was in real time scenario as well. [_\(M. Friedman, 1995\).](#)

Figure-8.5.1



[_\(user-acceptance-testing, n.d.\)](#)

The acceptance testing process has high chances to find out the bug which are missed by all three level of testing which are functional testing, Integration testing and system testing. The tester of customer who involved in the acceptance testing should have ability to think analytical way and logical way as well as they should have good knowledge about the domain and network. Moreover, they should have good understanding about the business need and how to test that requirement accordingly.

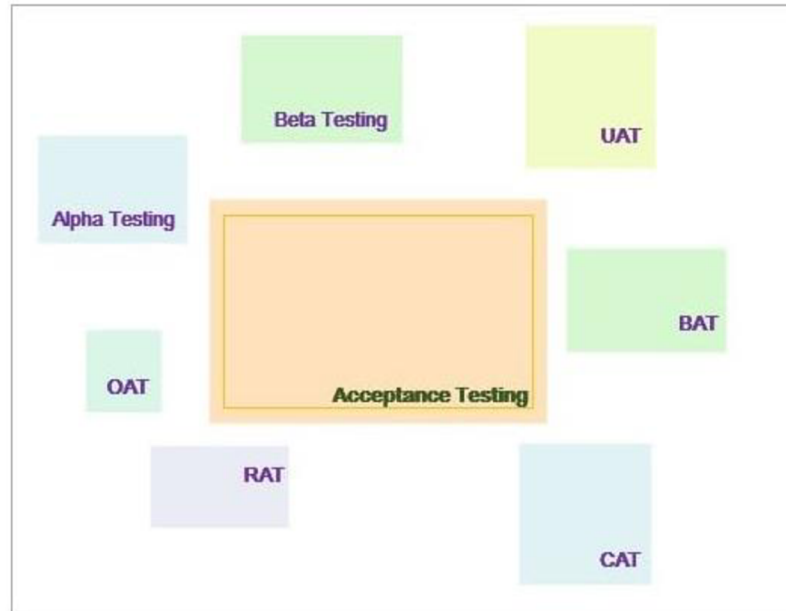
The Best practice is acceptance test plan should prepare while the test documents preparation for the testing phase in software development life cycle. The acceptance test plan should contain the information about the well-defined scope of acceptance testing and checklist before it performed.

The test plan document should be mentioned about the test step and test data to performed and check real scenario to set the clear expectations and test the end-to-end whole business flow work process.

There are several types of the acceptance testing are performed during this testing level which all testing technique are covering all the testing scope of acceptance test from the business side as well as functional and system operations side and they all are as per below. [_\(Testing, n.d.\)](#)

8.5.1 Types of acceptance testing:

Figure-8.5.1.1



_(Types-of-acceptance-testing, n.d.)

a) User acceptance testing (UAT):

User acceptance testing is also known as UAT testing. This is the most common testing part of the acceptance testing level. This is as the end-user testing. This is defined as the software testing which ensures the software product working as this is specified for the usage.

This testing is done usually by the end user so they can ensure the specific requirement which they quite often use and test it as per their point of view. The user acceptance testing also sometimes involves various end users across the country if their product is based and developed for the worldwide process. [_\(testing A. , n.d.\)](#)

b) Contract acceptance testing (CAT):

The contract acceptance testing is about the contract which specified and predetermined for the software product testing which the product will go live to use. During this period the software product should be passed through all acceptance use case and criteria.

Here contract signed as the service level agreement which means the payment will be made only in case of if the software product will full fill the whole requirement. This contract usually made before the software product goes live and the testing processes carried out as per the area of the testing. [\(M. Friedman, 1995\)](#).

c) Business acceptance testing (BAT):

Business acceptance testing is ensured and test the part of business requirements of software product and verify that the business requirement reaches its goals. Sometime the software product is passed and reached the goals, but they failed in the business acceptance criteria.

This testing main focused on the business-related financial criteria so in case of changes in requirement because of change in needs of market condition or due to advancing the technology the current implementation of the software product needs the additional budget. [\(testing A. , n.d.\)](#)

d) Regulation acceptance testing (RAT):

Regulation acceptance testing is type of software testing which insured the software product in term of the rule and regulation which is defined by the government of region or country where the software product will go for live for use. The software products are developed for the region or country or for worldwide.

The software product must go through the regulation acceptance testing to insure that it is not violating any rule or regulation which is define by the region or country. In case the software product is not developed or

violating the government rules somehow that county or region will not allow that software product and it will be considered as the failed in the regulation acceptance testing.

e) Operation acceptance testing (OAT):

The operation acceptance testing is type of non-function testing. Which is includes the non-functional testing criteria for the testing like reliability testing, compatibility testing, recovery testing, maintainability testing, fail over and globalization, localization, stability, scalability, recovery testing, performance testing and so many.

The operational acceptance testing mainly focuses on these non-functional aspects of the software product and make sure that software products performance is well enough and the product is stable enough to go for production. [\(M. Friedman, 1995\)](#).

f) Alpha testing:

Alpha testing is performed during when the product is under the test environment. It is carried out by the specialized testing team which are usually called the alpha tester. This phase mainly focused on the bug which are still uncover and some bugs which is creates the usability issue in the software product.

This tester also focused on the functionality and suggest about the feature gaps and fill it. After the alpha testing testers gives their suggestion and feedbacks which will help in improvement in the product functionality and performance as well as help in verifying and fixing the bugs. [\(M. Friedman, 1995\)](#).

g) Beta testing:

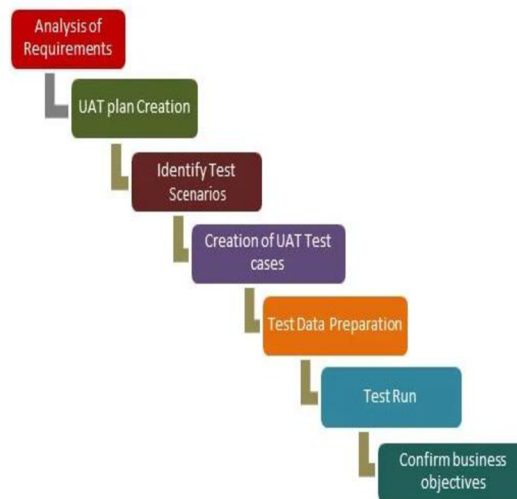
Beta testing is mostly done by the real end user which are called the beta tester the testing process is performed in their test environment for actual

use of software product over of some defined period and gain the experience. Beta testers ensure the product according to customer validation methodologies and give their recommendation and feedback on the performance, design, usability, UI and functionality which will help to improve the quality of product. Beta testing is performed before it goes to work in real market.

This testing usually performed on the various platforms to get the valuable suggestion and feedback from beta user or real user which ultimately result in successful beta testing of the software product and it is prove that client of customer is happy and satisfied with the developed software product and its usage. This is the best practice to test and verify the product before it goes to the live usage for the people. [_\(testing A. , n.d.\)](#)

8.5.2 Acceptance testing process workflow:

Figure-8.5.2.1



[_\(user-acceptance-testing, n.d.\)](#)

a) Analysis and gathering of requirements:

The analysis of requirement is the most important activity among all activities in acceptance testing. It helps to create and cover all the testing scope and develop test scenarios.

It will be useful for the develop the documentation like acceptance test plan, business requirement document, system requirement documents and specification, business process flow diagram and business and functional diagrams. [\(M. Friedman, 1995\)](#).

b) Acceptance test plan creation:

The acceptance test plan includes the proper plan how it will carry out the acceptance testing and strategy. The plan and strategy will help to properly test and verify that software product is developed as per the functional and business requirement, and it behave as it should be.

It includes the document like test scenario, test cases approach, scope of testing, entry and exit criteria for acceptance testing and period of testing.

c) Identifying the test scenario and testing test cases:

The function and business requirement are plays vital roles to understand the test scenario. Once the understanding is clear about the scenarios it will help to create test case and prepare the test data. The test cases should be cover most of the functionality according to all the system and business requirement of the software product. [\(testing A. , n.d.\)](#)

d) Preparing the test data:

The test scenario and test case should be prepared for the preparation of the test data. Also, the tester should have good knowledge and understanding about the database flows for test purpose. It is good practice to use the live data for the acceptance testing process. Data should be compactible.

e) Preparation for test run and document the result:

The test case and test data preparation should be done before the test execution run. Usually test case and data are prepared in test management

tool like Jira or test runner. With the use of test management tool we can get the exact statistic and graph about the test run process.

One test run has been done the result should be mentioned in the test management tool for the particular test case. It will help to create test run matrices like traceability matrices which can help in presentation to the team as well as the customer to show the current situation of test run. [\(M. Friedman, 1995\)](#)

f) Confirmation of business objectives met:

The confirmation about business objective meets and the software is developed and working as per the customer needs is send by the confirmation email for the sign off after the acceptance testing done. After this confirmation email about of sign off the software product will send for the production. [\(testing A. , n.d.\)](#)

8.5.3 Guidelines for the acceptance testing:

- 1) Acceptance testing becomes a more efficient and effective if the customer or real end user involved during the acceptance testing process.
- 2) Acceptance testing should be classified in Alpha and Beta testing process but if software product is not developed for service-based industries it does not matter. [\(Marick, 1995\)](#)
- 3) Acceptance testing should carry out always in Acceptance test environment.
- 4) Preparation of acceptance test plan should be done at early stage of software development life cycle.

- 5) Well define functional as well as business requirement are gathered and prepared.
- 6) Clear and defined set scope of acceptance testing and expectation.
- 7) Acceptance test scenario, test case, test data and test documents are prepared.
- 8) Database is updated for real world scenarios and data for acceptance test.
- 9) Tester or read end user should have analytical thinking and should think the software product as unknown user.
- 10) Tester should test end to end business flow.
- 11) Arrange a meeting for suggestion and feedback on acceptance test for the improvement before it move to production.
- 12) Prepare and verify the checklist before and after the acceptance test.

9. Practical Part

9.1 Unit testing:

Here now I am explaining you how unit test practically works in automation tool. As well as I have mentioned as comment in code and what they are executing as action in Amazon online shopping website. I have written the unit test scenario with test case sample as well as attaching selenium automation tool's screenshot with code to show you how it is executing the test case in automation as per given test case. The video link is uploaded on internet to see the actual execution of coding.

9.1.1 Test case sample for Unit testing:

Test Case ID	01.
Test Scenario	Login functionality Of Amazon.in online shopping website.
Test case title	Verify the log in functionality of Amazon.in online shopping website with valid login ID and password.
Pre-conditions	The account is already created on the Amazon.in website.
Dependencies	No, there is not any dependencies for this test case.
Test Designed By	Ankit Vaniya
Test Designed date	18/12/2020

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Open the website. Amazon.in in google chrome web browser.		Website is opened successfully.			
2	Click on sign in Button.		User is successfully navigated on the login page of Amzon.in website.			
3	Please enter the valid login id in the textbox.	e.g., ankit.xxxgm ail.com	Login id successfully added.			
4	Click on “continue” button.		Password textbox is appeared.			
5	Enter valid password in textbox.	e.g., Password: XXXXXXXXXX	Password is inserted successfully.			
6	Please click on sign in button.		The user is successfully login in amazon.in website.			

9.1.2 Screenshots of Unit testing:

Figure-9.1.2.1

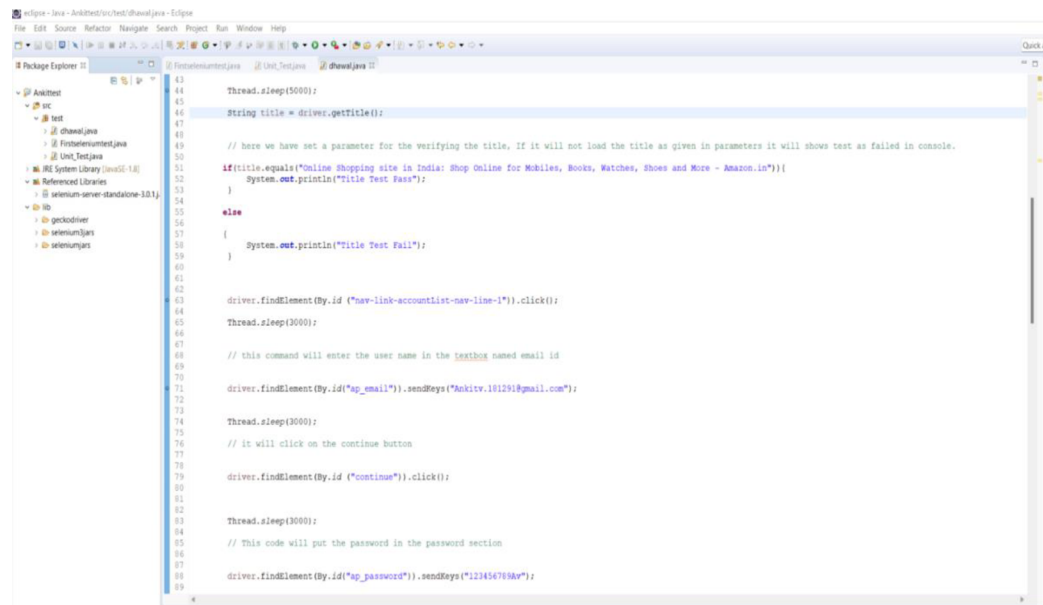
```

1 package test;
2 import java.util.ArrayList;
3 import java.util.List;
4 import org.openqa.selenium.support.ui.Select;
5 import org.openqa.selenium.By;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.WebElement;
8 import org.openqa.selenium.WebElement;
9 import org.openqa.selenium.chrome.ChromeDriver;
10 import org.openqa.selenium.support.ui.WebDriverWait;
11
12
13
14 public class @html {
15
16
17
18 // System Property for Google Chrome Driver
19
20 System.setProperty("webdriver.chrome.driver", "C:\\Users\\91814\\Downloads\\gecko\\chromedriver_win32\\chromedriver.exe");
21
22
23 // Instantiate a ChromeDriver class.
24
25
26 WebDriver driver=new ChromeDriver();
27
28 WebDriverWait wait = new WebDriverWait(driver,5000);
29
30 // For load the window in maximum size.
31
32 driver.manage().window().maximize();
33
34
35 //Searching for the amazon online shopping website and launching it
36
37 driver.get("https://amazon.in/");
38
39
40 // Thread . sleep is used as waiting time between executing two steps
41
42
43 Thread.sleep(5000);
44
45 String title = driver.getTitle();
46
47

```

Source: Author processing

Figure-9.1.2.2



```
43 Thread.sleep(5000);
44
45 String title = driver.getTitle();
46
47 // here we have set a parameter for the verifying the title, If it will not load the title as given in parameters it will shows test as failed in console.
48 if(title.equals("Online Shopping site in India: Shop Online for Mobiles, Books, Watches, Shoes and More - Amazon.in")){
49     System.out.println("Title Test Pass");
50 }
51 else
52 {
53     System.out.println("Title Test Fail");
54 }
55
56 driver.findElement(By.id ("nav-link-accountlist-nav-line-1")).click();
57 Thread.sleep(3000);
58
59 // this command will enter the user name in the textbox named email id
60 driver.findElement(By.id("ap_email")).sendKeys("Ankitv.181291@gmail.com");
61
62 Thread.sleep(3000);
63
64 // it will click on the continue button
65 driver.findElement(By.id ("continue")).click();
66
67 Thread.sleep(3000);
68
69 // This code will put the password in the password section
70 driver.findElement(By.id("ap_password")).sendKeys("123456789Aa");
71
72 Thread.sleep(3000);
73
74 // it will click on the submit button
75 driver.findElement(By.id ("signInSubmit")).click();
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

Source: Author processing

Figure-9.1.2.3

```
77
78
79 driver.findElement(By.id ("continue")).click();
80
81
82 Thread.sleep(3000);
83
84 // This code will put the password in the password section
85 driver.findElement(By.id("ap_password")).sendKeys("123456789Aa");
86
87 Thread.sleep(3000);
88
89 // it will click on the submit button
90 driver.findElement(By.id ("signInSubmit")).click();
91
92
93
94
95
96
97
98
99
```

Source: Author processing

9.1.3 Source codes for Unit testing:

```
package test;
import java.util.ArrayList;
import java.util.List;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Ankit {

    public static void main(String[] args) throws InterruptedException {
```

```

// System Property for Chrome Driver

System.setProperty("webdriver.chrome.driver",
"C:\\Users\\91814\\Downloads\\gecko\\chromedriver_win32\\chromedriver.exe");

// Instantiate a ChromeDriver class.
WebDriver driver=new ChromeDriver();

WebDriverWait wait = new WebDriverWait(driver,5000);

// for load the window in maximum size.
driver.manage().window().maximize();

//Searching for the amazon online shopping website and launching it
driver.get("https://amazon.in/");

// Thread . sleep is used as waiting time between executing two steps
Thread.sleep(5000);

String title = driver.getTitle();

// here we have set a parameter for the verifying the title,
// If it will not load the title as given in parameters, it
// will shows test as failed in console.

if(title.equals("Online Shopping site in India: Shop Online for Mobiles,
Books, Watches, Shoes and More - Amazon.in")){

    System.out.println("Title Test Pass");
}

else

{
    System.out.println("Title Test Fail");
}

driver.findElement(By.id ("nav-link-accountList-nav-line-1")).click();

Thread.sleep(3000);

// this command will enter the username in the textbox named email id

driver.findElement(By.id("ap_email")).sendKeys ("Ankitv.xxxxx@gmail.com");

```

```

Thread.sleep(3000);

// it will click on the continue button

driver.findElement(By.id ("continue")).click();

Thread.sleep(3000);

// This code will put the password in the password section

driver.findElement(By.id("ap_password")).sendKeys("xxxxxxxxxAv");

Thread.sleep(3000);

driver.findElement(By.id ("signInSubmit")).click();

```

9.2 Integration testing:

Here now I am explaining you how Integration test practically works in automation tool. As well as I have mentioned as comment in code and what they are executing as action in Amazon online shopping website. I have written the integration test scenario with test case sample as well as attaching selenium automation tool's screenshot with code to show you how it is executing the test case in automation as per given test case. The integration testing practical part is related to unite test so please consider its screenshots and coding from the automation tool. The video link is uploaded on internet to see the actual execution of coding.

9.2.1 Test case sample for Integration testing:

Test Case ID	02.
Test Scenario	Search bar functionality of Amazon.in website.
Test case title	Verify the search button is working properly with database of Amazon.in online shopping website.
Pre-conditions	The account is already created on the Amazon.in website.
Dependencies	No, there is not any dependencies for this test case.
Test Designed By	Ankit Vaniya
Test Designed date	18/12/2020

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Please perform all steps from test case 01.		All the steps from the test case 01 is successfully performed.			
2	Please check the user is successfully logged in the Amazon.in website.		The User is successfully logged on the website.			
3	Please Enter the data in search bar and click to search.	Data: I-phone13 256 white.	The Search result for the I-phone13 product is visible on the webpage.			
4	Click on I-phone as per your choice.		I-phone is selected and opened in new tab.			
5	Select the quality, color etc. of your choice.		Options are selected for the chosen I-phone product.			
6	Please check if “Add to cart” button visible.		“Add to cart button” is visible on screen.			
7	Please click on that button.		The product is successfully added to the cart.			

9.2.2 Screenshots of Integration testing:

Figure-9.2.2.1

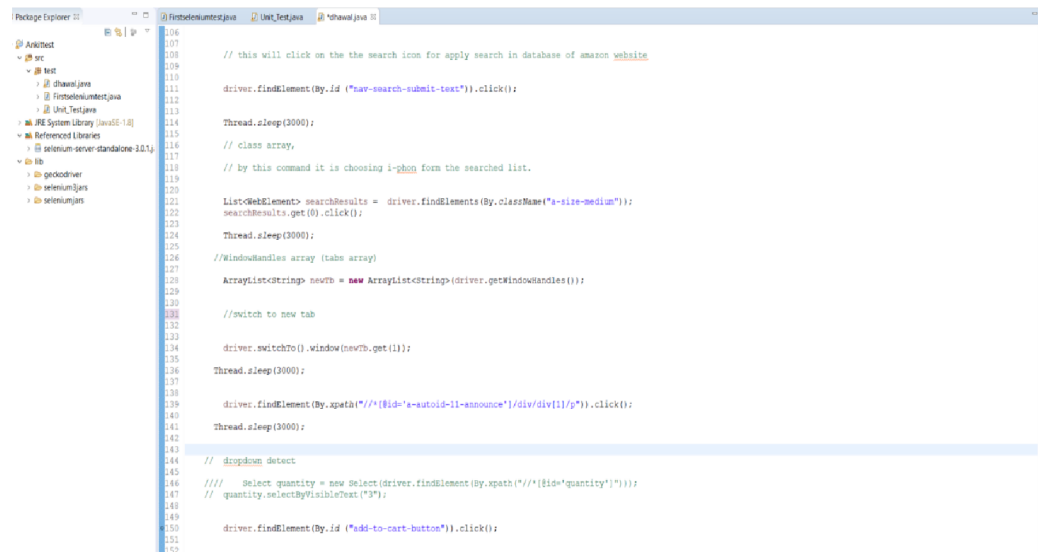
```

101
102 // this will enter the i-phone model name and number in the search bar
103
104
105 driver.findElement(By.id ("twotabsearchtextbox")).sendKeys ("iphone 13 256 white");
106
107 // this will click on the the search icon for apply search in database of amazon website
108
109 driver.findElement (By.id ("nav-search-submit-text")).click ();
110
111 Thread.sleep (3000);
112 // class array,
113 // by this command it is choosing i-phon from the searched list.
114
115 List<WebElement> searchResults = driver.findElements (By.className ("a-size-medium"));
116 searchResults.get (0).click ();
117
118 Thread.sleep (3000);
119 //WindowHandles array (tabe array)
120
121 ArrayList<String> newTab = new ArrayList<String> (driver.getWindowHandles ());
122
123 //switch to new tab
124
125 driver.switchTo ().window (newTab.get (1));
126
127 Thread.sleep (3000);
128
129 driver.findElement (By.xpath ("//*[@id='a-autoid-11-announce']/div/div[1]/p")).click ();
130
131 Thread.sleep (3000);
132
133 // dropdown detect
134
135 // // Select quantity = new Select (driver.findElement (By.xpath ("//*[@id='quantity']")));

```

Source: Author processing

Figure-9.2.2.2



The screenshot shows an IDE window with a Package Explorer on the left and a code editor on the right. The Package Explorer shows a project structure with folders for 'src', 'test', and 'resources'. The code editor displays Java code for Selenium testing. The code includes comments in green and code in black. The code performs the following actions: 1. Clicks on the search icon. 2. Waits for 3000ms. 3. Finds a class array. 4. Clicks on the 'Iphone' item from the search results. 5. Waits for 3000ms. 6. Finds a list of elements with class 'a-size-medium'. 7. Clicks on the first element. 8. Waits for 3000ms. 9. Finds a new tab array. 10. Switches to the new tab. 11. Waits for 3000ms. 12. Clicks on the 'add-to-cart-button'.

Source: Author processing

9.2.3 Source codes for Integration testing:

```
// this will enter the Iphone model name and number in the
search bar

driver.findElement(By.id
("twotabsearchtextbox")).sendKeys("Iphone 13 256 white");

// this will click on the the search icon for apply search in
database of amazon website

driver.findElement(By.id ("nav-search-submit-text")).click();

Thread.sleep(3000);

// class array,

// by this command it is choosing Iphone form the searched list.

List<WebElement> searchResults =
driver.findElements(By.className("a-size-medium"));
searchResults.get(0).click();

Thread.sleep(3000);

//WindowHandles array (tabs array)
```



```

        ArrayList<String> newTb = new
ArrayList<String>(driver.getWindowHandles());

        //switch to new tab

        driver.switchTo().window(newTb.get(1));

        Thread.sleep(3000);

        driver.findElement(By.xpath("//*[@id='a-autoid-11-
announce']/div/div[1]/p")).click();

        Thread.sleep(3000);

        driver.findElement(By.id ("add-to-cart-button")).click();

```

9.3 System testing:

Here now I am explaining you how system test practically works in automation tool. As well as I have mentioned as comment in code and what they are executing as action in Amazon online shopping website. I have written the system test scenario with test case sample as well as attaching selenium automation tool's screenshot with code to show you how it is executing the test case in automation as per given test case. The system testing practical part is related to unite test and integration test so please consider its screenshots and coding from the automation tool. The video link is uploaded on internet to see the actual execution of coding.

9.3.1 Test case sample for System testing:

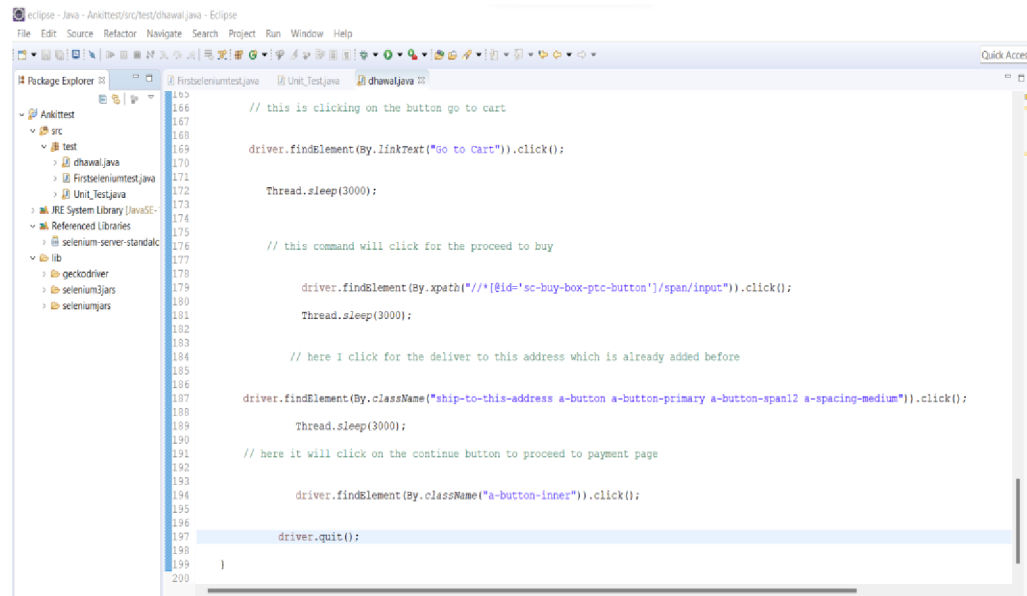
Test Case ID	03.
Test Scenario	Amazon.in system process from searching product to payment.
Test case title	Verify the system process flow of Amazon.in website from login, searching to payment process.
Pre-conditions	The account is already created on the Amazon.in website.
Dependencies	No, there is not any dependencies for this test case.

Test Designed By	Ankit Vaniya
Test Designed date	18/12/2020

step	Test steps	Test data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Please perform all steps from test case 01.		All the steps from the test case 01 is successfully performed.			
2	Please check the user is successfully logged in the Amazon.in website		User is successfully logged in.			
3	Please Enter the data in search bar and click on search.	Data: I-phone13 256 white.	Search result for the I-phone product is visible in webpage.			
4	Click on I-phone as per your choice.		I-phone is selected and opened in new tab.			
5	Select the quality, color etc. of your choice.		Options are selected for the chosen product.			
6	Please check if “Add to cart” button visible.		“Add to cart button” is visible on screen.			
7	Please click on that button.		The product is successfully added to the cart.			
8	Please click on proceed to checkout button		User is navigated to shipping address page.			
9	Please select the address which is already exist there.		The shipping address is selected and continue button visible now.			
10	Please click on continue button		User is navigated to payment option page.			

9.3.2 Screenshots of System testing:

Figure-9.3.2.1



Source: Author processing

9.3.3 Source codes for System testing:

```
// this is clicking on the button go to cart

driver.findElement(By.linkText("Go to Cart")).click();

Thread.sleep(3000);

// this command will click for the proceed to buy

driver.findElement(By.xpath("//*[@id='sc-buy-box-ptc-button']/span/input")).click();

Thread.sleep(3000);

// here I click for the deliver to this address which is
already added before

driver.findElement(By.className("ship-to-this-address a-
button a-button-primary a-button-span12 a-spacing-medium")).click();

Thread.sleep(3000);

// here it will click on the continue button to
```

```

        proceed to payment page.

        driver.findElement(By.className("a-button-
inner")).click();

        driver.quit();

    }
}

```

9.4 Test case execution video link:

https://docs.google.com/presentation/d/1Z9qA3BAP517iJjVuMn1U_gpWP-EEVRP/edit?usp=share_link&ouid=102686019584461517141&rtpof=true&sd=true.

9.5 Approximate data related the time/cost/factors of manual and automation testing:

9.5.1 Approximate time taken for Test case/ Script preparation:

Tester	Experience	Test case 1 Preparation time.	Test case 2 Preparation time.	Test case 3 Preparation time.
Manual Tester 1	6m – 1 year	20 – 22 Minutes	18 – 20 Minutes	15 – 17 Minutes
Manual Tester 2	2.5 years	15 – 18 Minutes	13 – 15 Minutes	7 - 10 Minutes
Manual Tester 3	3 years +	10 – 12 Minutes	8 – 10 Minutes	5 - 7 Minutes
End User	-	15 – 30 Minutes	13 – 20 Minutes	10 – 15 Minutes
Automation Tester 1	6m – 1 year	40 – 1 hour	30 – 40 Minutes	20 -25 Minutes
Automation Tester 2	3.5 years	10- 15 Minutes	17 - 20 Minutes	22 – 25 Minutes

9.5.2 Manual tester's approximate test case execution time with hourly/monthly wages:

Tester	Experience	Test case 1 Execution time.	Test case 2 Execution time.	Test case 3 Execution Time.	Approximate Wages per Hour/month
Manual Tester 1	6m – 1 year	1.10 Minutes	1.55 Minutes	2.35 Minutes	255.68 / 45000 CZK
Manual Tester 2	2.5 years	55 Seconds	1.40 Minutes	2.10 Minutes	304.97 / 53675 CZK
Manual Tester 3	3 years +	50 seconds	1.30 Minutes	1.50 Minutes	505.59 / 88984 CZK
End User	-	1.35 Minutes	2.10 Minutes	2.55 Minutes	-

9.5.3 Automation tester's approximate test case execution time with hourly/monthly wages:

Automation tester	Experience	Test case 1 Execution time.	Test case 2 Execution time.	Test case 3 Execution Time.	Approximate wages Hour/month
Automation Tester 1	6m – 1 year	31 sec	46 sec	52 sec	323.56 / 57000 CZK
Automation Tester 2	3.5 years	18 sec	27 sec	37 sec	547.23 / 96131 CZK

9.5.4 Comparison of manual and automation testing as per the software testing factors:

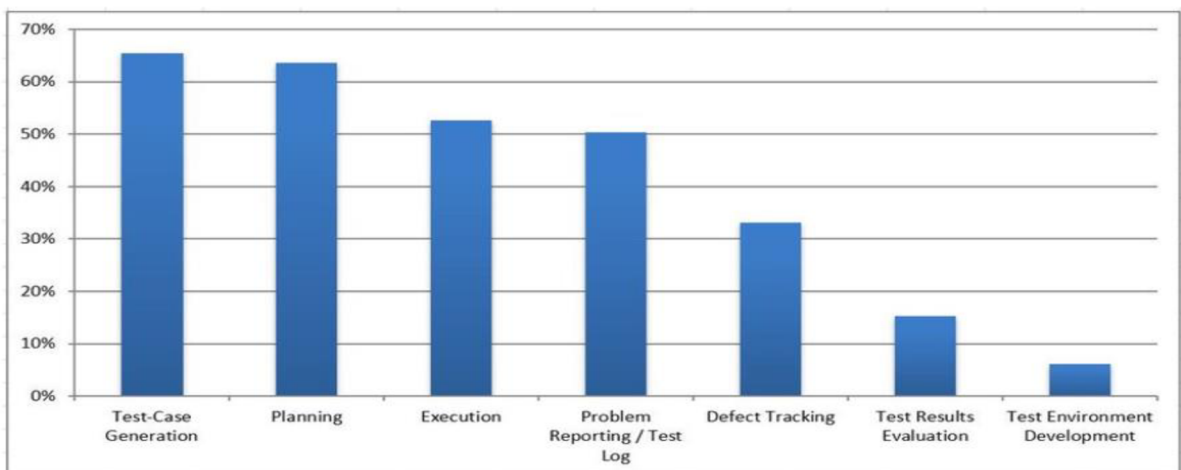
I have shown you the in previous tables the Manual testers and Automation tester's approximate time consuming for the test case preparation and test case execution as per their individual Manual and Automation tester's experiences. Also, I mentioned the Individuals hourly and monthly wages as per experience. As per that I have figured out about some of the factors of testing and which testing is better for which factor is as per below.

Factors	Manual testing	Automation testing
Human efforts	More	Less
Time	More	Less
Money	Cheap for small project	Cheap for long big project
Reusability	Less	More
Programing Knowledge	No	Yes
Accuracy	Less	High
Customer experience	Help in improving the customer experience	No guarantee for the positive customer experience
Compatibility	High	Less

9.6 Economic benefits of Software quality assurance, Manual testing and Automation testing:

First, as these are the most important phases of software development life cycle can make the direct impact on the economic benefits on the project. The software development without this phase can cost a lot of money and time. Economic benefits also depend on the software development model followed while software development. As well the economic benefits also depend on the complexity of project. Moreover, the software development team members and their role and experience are also matter for the economic benefits.

Role	Responsibilities	Average salary in Prague
Test Manager	Manager all activity, Negotiation with management, development and customer, Preparation of test approach and strategy.	464.94 Czech Koruna per hour.
Test Leader	Reporting and negotiation with test manager, creation of test plan according to test strategy, creating task for team member according to requirement, fulfill the requirement for testing.	411.93 Czech Koruna per hour.
Test Analyst	Analysis and preparation for test documentation.	281.94 Czech Koruna per hour.
QA Engineer	Implementation of defined software quality assurance processes and standard.	330.13 Czech Koruna per hour.
Software Manual Tester	Testing software manually, creation test cases, test execution report, bug report, retesting.	276.70 Czech Koruna per hour.
Software Automation Tester	Managing and creating automation testing. Creation test report and documentation.	378.78 Czech Koruna per hour.



Distribution of job for the testing tasks.

If the testing related phases would be avoided or not involved at right time it could cost a lot of effort, money, and time for fixing later. It could be also matter of the software development company's reputation in the market. The software development model which are involving the testing phases parallel same as development would make the drastic change on the economic situation of software development project.

The complexity of the project is also making impact on the economic benefits of the software development project. If the project is too big and need more resources and have enough budget than it is better idea to outsource the labour work. If we can check the table below and compare the prices according to international currency exchange rate and we can simply understand that if we outsource the workload which can make large impact on economic situation of project.

Country	Software Quality Assurance	Software testing	Automation testing	Approx exchange rate
Czech Republic	58104 Czech Koruna / M	48700 Czech Koruna / M	66666 Czech Koruna / M	1 CZK = 0.045 USD and 3.5 INR.
India	45000 – 50000 Indian Rupee / M	24710 – 26129 Indian Rupee / M	33481 – 34742 Indian Rupee / M	1 INR = 0.012 USD, 0.26 CZK
USA	6406.91 USD / M	5719.16 USD / M	7155.75 USD / M	1 USD = 22 CZK, 82.10 INR.

9.7 Why Selenium tool is good for Automation testing?

Economic Benefits also depend on the automation tool used during the testing phase. This thesis purpose I used the selenium automation tool as it was not paid and open-source software which mean its free available on internet without any cost. Also, to learn and understand this software there is lots of tutorials and materials available on






YouTube and Google and it is also having free access. Here I have a screenshot which is showing the serve regarding the automation tool using by the companies and the highly used automation tool is Selenium automation tool which is 43.2% compared to the automation tool. The main benefits of selenium automation tool are we can create multiple Framework with multiple programming language for multiple browsers.

Selenium	Automated test execution	43.2%
Jira	Test management, bug tracking	38%
Cucumber	Automated test execution	9.5%
SoapUI	Automated test execution, API testing	8.5%
Jmeter	Performance Testing	7.8%
Team Foundation Server	Test management	7.2%
HP QC / ALM	Test management	6.3%
LoadRunner	Performance Testing	6.2%
Confluence	Test management	5.8%
TestNG	Automated test execution, unit testing	5.8%
HP QTP	Automated test execution	5.5%
TestRail	Test management	5.5%
JUnit	Unit testing	4.4%
Katalon Studio	Automated test execution	2.65%
Bugzilla	Bug tracking	1.94%
TestComplete	Automated test execution	1.94%
Microsoft Test Manager	Test management	1.76%
Ranorex	Automated test execution	1.41%
Tosca	Automated test execution	1.23%

Language	Platforms	Browser
Java	Mac	Google Chrome
Java Script	Windows	Firefox
Python	Linux	Safari
PHP	Android (Selendroid, Appium, Robotium)	Opera
Ruby	iOS (with ios-driver or appium)	Microsoft Edge
C#		Internet Explorer
Objective-C		

Selenium compatibility with Language, platform and browsers.

Figure-9.7.1

Product	 Selenium	 Katalon Studio	 Unified Functional Testing	 TestComplete	 watir
Available since	2004	2015	1998	1999	2008
Application Under Test	Web apps	Web (UI & API), Mobile apps	Web (UI & API), Mobile, Desktop, Packaged apps	Web (UI & API), Mobile, Desktop apps	Web apps
Pricing	Free	Free	\$\$\$\$	\$\$	Free
Supported Platforms	Windows Linux OS X	Windows Linux OS X	Windows	Windows	Windows Linux OS X
Scripting languages	Java, C#, Perl, Python, JavaScript, Ruby, PHP	Java/Groovy	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ and C#	Ruby
Programming skills	Advanced skills needed to integrate various tools	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Advanced skills needed to integrate various tools
Ease of Installation and Use	Require advanced skills to install and use	Easy to setup and use	Complex in installation. Need training to properly use the tool	Easy to setup. Need training to properly use the tool	Advanced skills needed to integrate various tools

_(Google Images, n.d.)

Comparison of Selenium with other automation tools

10. Conclusion

This thesis introduced with the brief information about software quality assurance and software testing process. As the software testing is one of the most important phases of software development life cycle and it is making significant impact on the matter of software product quality and project economic. I have mentioned and described the process and difference between quality assurance and testing processes and how these techniques are used to measure quality of software product. Moreover, this thesis consisted of a practical part showing manual and automation testing sample how testing levels are proceeded testing processes after one by one to provide and maintain best quality of software product. This thesis also proving the better testing type as per the factor with the comparison of tester experience, time taken, and money investment. Thank you so much.

11. Bibliography

Brian Marick, *The Craft of Software Testing*, Prentice Hall, 1995.

Kaner, C. (1996). *Software negligence & testing coverage*

Michael Dyer, *The Cleanroom Approach to Quality Software Development*, Wiley, 1992.

M. Friedman and J. Voas, *Software Assessment: Reliability, Safety, Testability*, Wiley, 1995

Srinivasan Desikan, *Software Testing: Principles and Practice*, 2006

Ankit Vaniya, *CZU Diploma Thesis, Software Testing*, 2018

<https://www.softwaretestinghelp.com/software-quality-assurance/>

<https://www.guru99.com/all-about-quality-assurance.html>

<https://www.javatpoint.com/quality-assurance>

<https://www.softwaretestpro.com/wp-content/uploads/2016/10/AutomatedTestingHandbook.pdf>

<https://stackify.com/ultimate-guide-performance-testing-and-software-testing/>

<https://www.javatpoint.com/levels-of-testing>

https://www.tutorialspoint.com/software_testing/software_testing_levels.htm

<https://www.softwaretestinghelp.com/system-testing/>

<https://www.softwaretestinghelp.com/what-is-acceptance-testing/>

<http://desy.lecturer.pens.ac.id/Manajemen%20Kualitas%20Perangkat%20Lunak/ebook/Software%20Quality%20Assurance%20From%20Theory%20to%20Implementation.pdf>

<https://www.geeksforgeeks.org/software-engineering-software-quality-assurance/>

<https://www.softwaretestpro.com/wp-content/uploads/2016/10/AutomatedTestingHandbook.pdf>

<http://tisten.ir/wp-content/uploads/2019/01/Complete-Guide-to-Test-Automation-Techniques-Practices-and-Patterns-for-Building-and-Maintaining-Effective-Software-Projects-Apress-2018-Arnon-Axelrod.pdf>

<https://www.guru99.com/automation-testing.html>

<https://testinium.com/blog/top-10-benefits-of-automated-testing-2/>

<https://jelvix.com/blog/software-testing-life-cycle>

<https://www.softwaretestinghelp.com/automation-testing-tutorial-1/#:~:text=Automation%20testing%20is%20a%20Software,are%20difficult%20to%20perform%20manually.>

https://www.getxray.app/blog/the-complete-guide-to-test-automation-best-strategies-frameworks-and-tools?utm_term=&utm_campaign=DSA+Search+-+Xray+WW+-+Blog+Posts&utm_source=adwords&utm_medium=ppc&hsa_acc=9970092548&hsa_cam=13952027914&hsa_grp=121805042381&hsa_ad=535183409030&hsa_src=g&hsa_tgt=aud-1243128720715:dsa-1392437111708&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQjw8_qRBhCXARIsAE2AtRY8o3ZF3CZh_aiMUIVttII2XymaBtRXcuoE4--VhnyVpv9tTT0HmAUAAtAZEALw_wcB

<https://www.google.co.in/>

<https://www.javatpoint.com/automation-testing>

<https://www.softwaretestinghelp.com/what-is-acceptance-testing/>

<https://www.guru99.com/test-automation-framework.html>

<https://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/>

<https://www.guru99.com/performance-testing.html>

<https://blog.testlodge.com/what-is-user-acceptance-testing/>

<https://www.geeksforgeeks.org/performance-testing-software-testing/#:~:text=Performance%20Testing%20is%20a%20type,stability%20under%20a%20particular%20workload.>

<https://www.microfocus.com/en-us/what-is/performance-testing>

https://www.tutorialspoint.com/software_testing_dictionary/performance_testing.htm

<https://www.softwaretestinghelp.com/introduction-to-performance-testing-loadrunner-training-tutorial-part-1/>

<https://stackify.com/ultimate-guide-performance-testing-and-software-testing/>

<https://www.softwaretestingclass.com/wp-content/uploads/2016/06/Beginner-Guide-To-Software-Testing.pdf>

<https://www.quora.com/What-is-acceptance-testing-and-what-are-the-types-of-acceptance-testing-Who-is-involved-in-acceptance-testing>

<https://www.getxray.app/blog/6-most-popular-test-automation-frameworks>

<https://www.guru99.com/levels-of-testing.html>

<https://u-tor.com/topic/software-testing-levels>

https://www.tutorialspoint.com/software_testing/software_testing_levels.htm

<https://www.javatpoint.com/levels-of-testing>

<https://www.geeksforgeeks.org/levels-of-software-testing/>

<https://www.softwaretestinghelp.com/types-of-software-testing/>

<https://softwaretestingfundamentals.com/system-testing/>