**Ques - Tell me about yourself?**

My name is Rahul Patidar, I am from Indore Madhya Pradesh India. I completed my B.E. in 2014 and after that I started my career in IT industry. I have a total 2.7 yr. of experience in software testing and 1+ year in automation testing.

Have a working knowledge in Waterfall and in Agile methodology, worked on project management tool Jira and application management tool ALM (Application management lifecycle).

In my previous project I performed System Testing, functional testing, Regression testing, UAT testing and sanity testing for different different products on different different channels in different different environment.

Working knowledge on requirement analysis, Written test cases for the application module and written automation script using Selenium WebDriver and TestNG and executed the same.

Created DDF framework and Hybrid framework in my project.

Coming to my hobbies I love to play badminton, chess and also enjoy to watch comedy and dance shows.

Do you want me to explain about my project?

If Yes- explain the project completely.

**Ques - Tell me about your project? What challenges you faced? how you handled? What was your role in a project?**

Actually I worked in a 2 project will explain one by one- in my 1st project I worked for insurance client who sells the insurance for commercial products like- properties, vehicles, farmweb related products, hotel pub etc. and for that one we performed Pricing testing, UW rule testing and documents testing.

We were getting the document from the client that is BRD, 1st will analysis that document and mark what is the business requirement will divide the requirements into modules and based on that will write the test cases (either in Excel or in ALM).

For UW rule and document testing we performed manually but for Pricing testing we created data driven framework because in this one we need to test multiple set of combination in a single application so we automate this using Selenium WebDriver. And also for all the products we need to test our pricing test in all the given channels that is why we created DDF so we can use our script for all channels and all the products. And for the same we followed ATLC model.

**ATLC (automation test life cycle)-**

**1.Automation feasibility analysis**-  Which test case can be automated and how we can automate them. Also will check which test cases cannot be automate.

**2.Test plan and Design**- Fetch all the manual test case from test management tool that which TC has to automate. Which framework to use and what will be advantage and disadvantage of the framework which we will use.

**3.enviorment setup**-  What should be the configuration in terms of hardware and software.

**4.test script creation-**

1.Start creating test script based on your requirement

2- Create some common method or function that you can reuse throughout your script

3- Make your script easy, reusable, well-structured and well documented so if third person check your script then he/she can understand your scripts easily.

4- Use better reporting so in case of failing you can trace your code

5- Finally review your script and your script should be ready before consumption.

**5.Test case execution**

1-Your script should cover all the functional requirement as per test case.

2-  Your script should be stable so it should run in multiple environment and multiple browsers (depends on your requirement)

3-   You can do batch execution also if possible so it will save time and effort.

4-   In case of failure your script should take screen shots.

5- If test case is failing due to functionality, you have to raise a bug/defect.

**6. Analysis of result**

1. Analyze the output and calculate how much time it takes to complete the test case.

2- You should have good report generation like XSLT report, TestNG report, ReporterNG etc.

In my 2nd project I tested different different modules (home page, login page, my work, location or documentation etc.) for an application in a different different environment so for that we created Hybrid framework.

We used Excel sheet as a data source where all the environment links and their credential is stored, test cases as also written in the excel sheet so after testing we update the results in an excel sheet.

We created packages like- Pages (all the page class), test case (all the test case), factory (Browser factory and dataprovidefactory class), utility (reusable classes)

Folders- application data, screenshots, reports etc.

1. Synchronization is big problem is automation but for that one we will use Implicit and explicit wait but sometimes you will not get or locate proper element so that time you need to write some logic- like refresh or run a loop 2-3 times so maybe you will get.
2. Finding locator is sometimes difficult on dynamic pages.
3. When you are designing any framework that time you need to take care of Base class, all the reusable logics (screenshot, reporting, logging), handling developer mode extension or authentication problem. Means maintenance of framework is bit difficult.
4. Dealing and handling with Exception is also tough challenge.
5. When you are integrating selenium with TestNG or Maven need to take care of all the libraries and dependencies.

**Ques - What is Framework, which framework you used in your project and why, explain architecture and modules/ components of framework?**

Framework means set of rules, guidelines, concepts, practices that we need to follow is our automation script. And with the help of framework we can design clean, robust and string script.

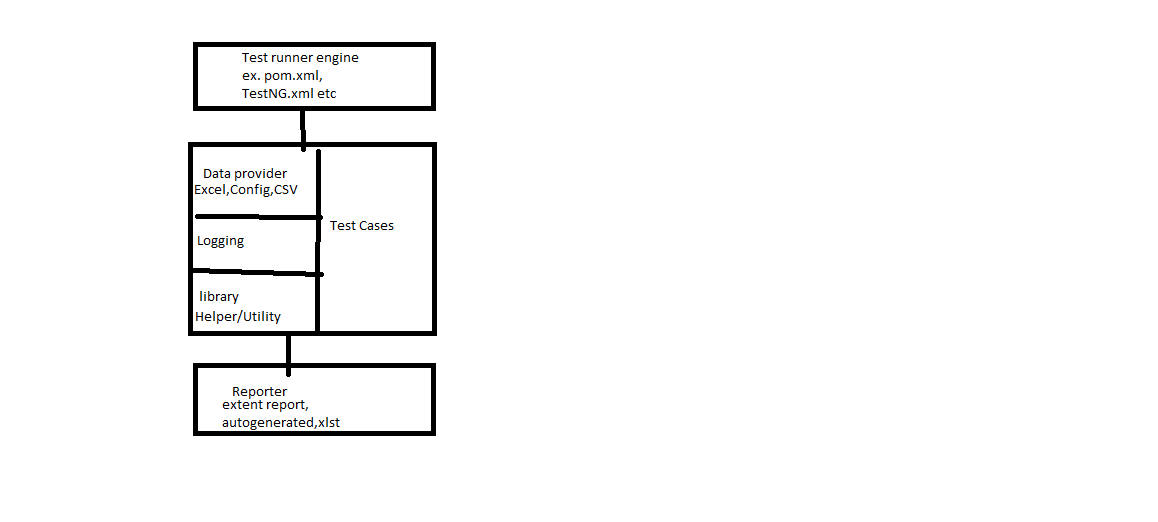
There is no defined structure for framework, each company is using their own way to design framework.

And for the designing of framework we used TestNG, POM concept so that will reuse our script.

I created data driven and hybrid framework in my project. And the reason is in my project we were getting the multiple set of data in excel sheet and that data we need to test in a same application for the pricing testing so we used DDF.

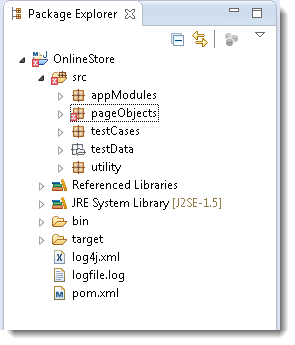
An also I design Hybrid framework modules- actually we were testing full application where number of modules were there like –Home page, search page, login page, My work , documentation, Location etc. and for the navigate and login scenario we used to fetch the data from excel sheet and also test cases is available in the excel so need to verify in the application by writing script and update the status in the sheet so we used Hybrid framework

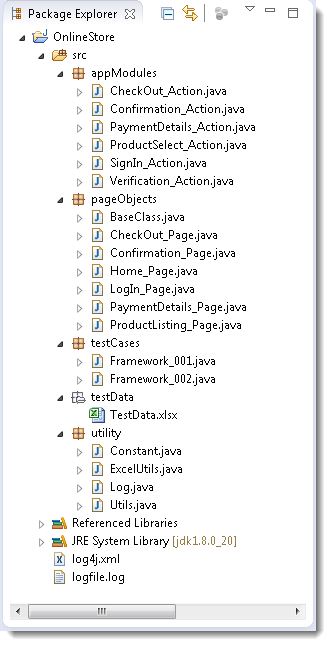
First I will explain the architecture of framework.



Data provider can be- Run data provider (Configuration, URL, Wait) ot may be Test data provider(Excel, xml etc)

Folder Structure-





Components/Modules-

* 1. Test Case Standarization (all the test case should follow same approach)
  2. Logging Standarization (common way)
  3. Test data & Configuration Utility (Excel, confi file)
  4. Helper/Utility Library (common functionality- Screenshot, login etc)
  5. Test execution engine (POM, TestNG etc)
  6. Reporting Utility.

**Ques - Why do you want to work here?**

According to me changes are necessary for everyone to enhance your skills, knowledge and for personal and financial growth. And this is the good platform where I can learn and gain more.

**Ques - What are your strength?**

I can easily adjust in new situation, new environment with new people I can say am adaptive by nature.

Flexibility- can work in any team

Persistence –complete our work in given deadline.

Love to explore.

**Ques - Weakness?**

Sometimes feel bored doing same thing again and again.

Bit lazy about which I am not interested like submit a timesheet, give feedback etc.

**Ques - Why should I hire you?**

Company is looking for the selenium profile and with reference to my experience, I satisfy all the requirement for this job, am sincere with my work and would never let you down in anyway. I promise you will never regret for the decision to appoint me in your organization.

**Ques - Tell me about this company?**

**Ques - Why you are looking for a job change?**

Relate to career goal am thankful to my previous organization because I have learnt a lot of things from there. According to me changes are necessary for everyone to enhance your skills, knowledge and for personal and financial growth. And this is the good platform where I can learn and gain more.

**Ques - What are your salary requirement?**

**Ques - What are your career goal?**

**Ques - Do you have any question to ask me?**  
Express thanks

**Ques - What you mention in Status mail?**

Whatever work I completed including Test case writing, execution their status and risks.

What are the issues with their Risks?

Defects and the reason

**Ques- Verification and Validation ?**

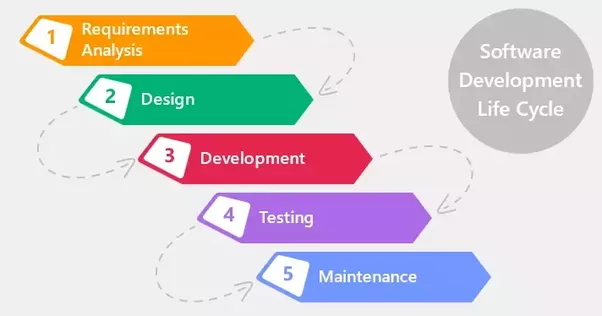
**Ques – Agile and Waterfall?**

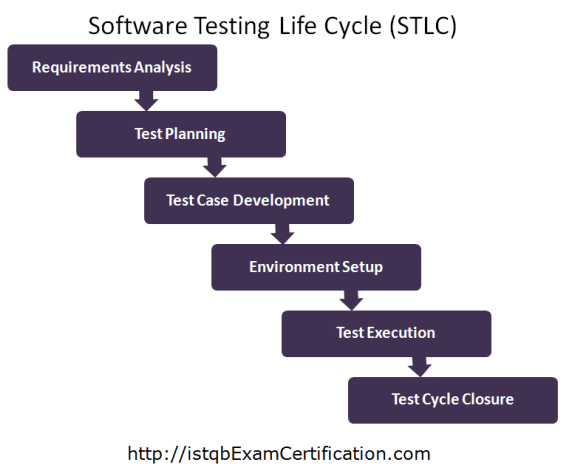
**Ques - How often your scrum works?**

**Ques - What is test estimation, planning, strategy and the difference?**

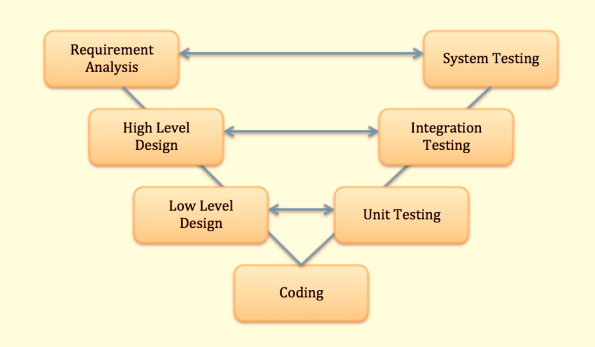
**Ques - Diff b/w test scenario and test cases?**

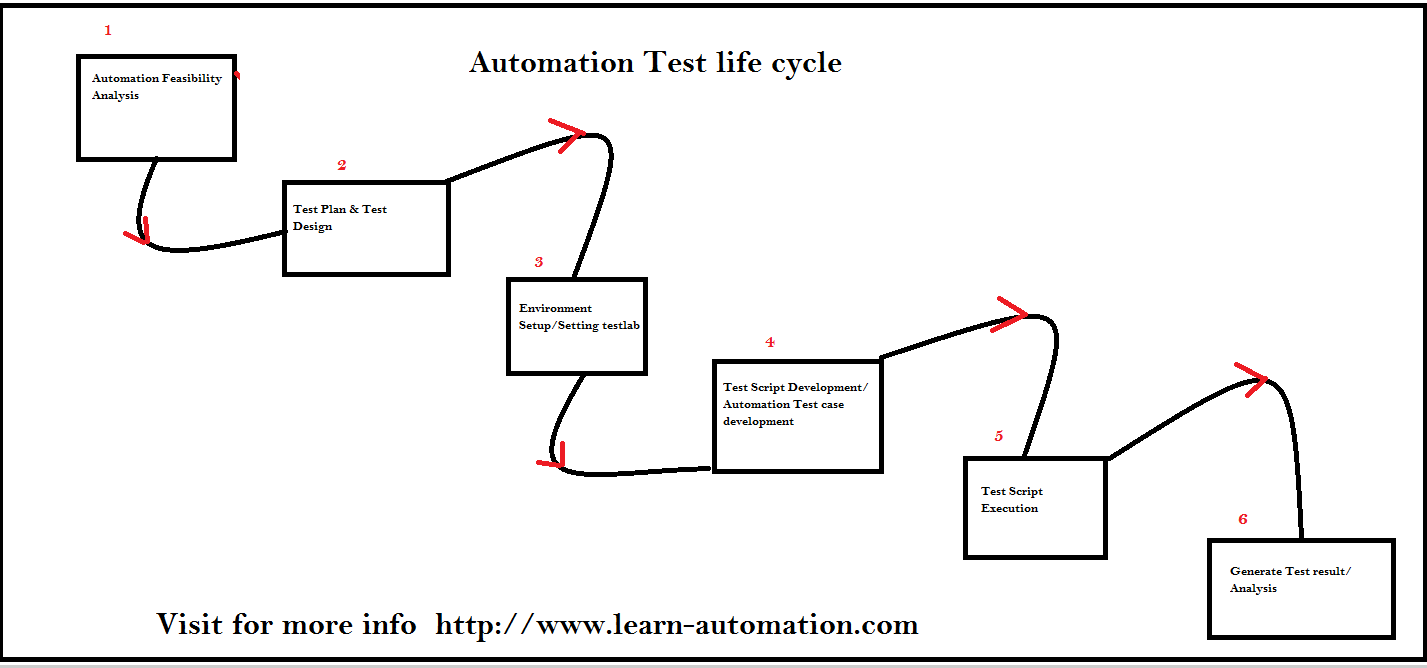
**Ques - What is STLC, SDLC, ATLC and explain?**



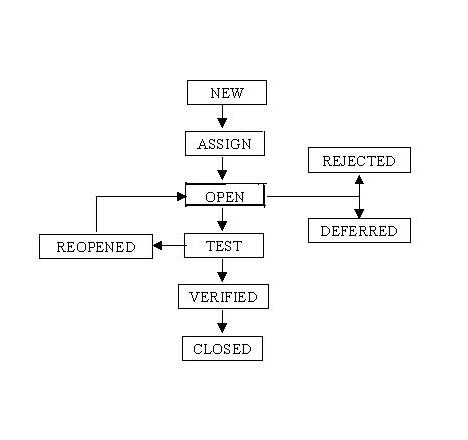


**V-Model**





**Ques - Defect/Bug life cycle?**



**Ques - RTM (Requirement traceability matrix)?**

**Ques - Defect management tool – ALM/QC explain their components?**

HP ALM (Application Life Cycle Management) is a web based tool that helps organizations to manage the application lifecycle right from project planning, requirements gathering, until Testing & deployment, which otherwise is a time-consuming task. supports various phases of the software development life cycle.

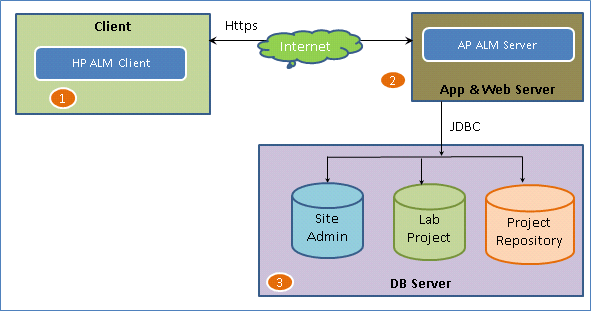
The various stakeholders involved in a typical project are –

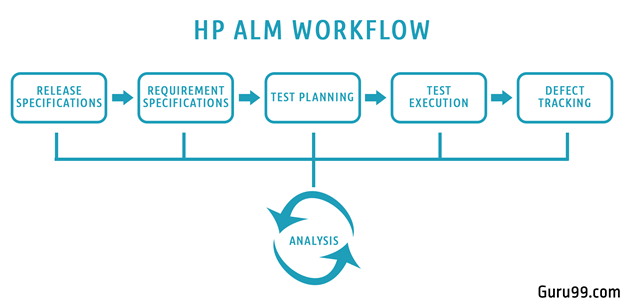
* Developer
* Tester
* Business Analysts
* Project Managers
* Product Owners

These stakeholders perform diverse set of activities that need to be communicated to all concerned team members.

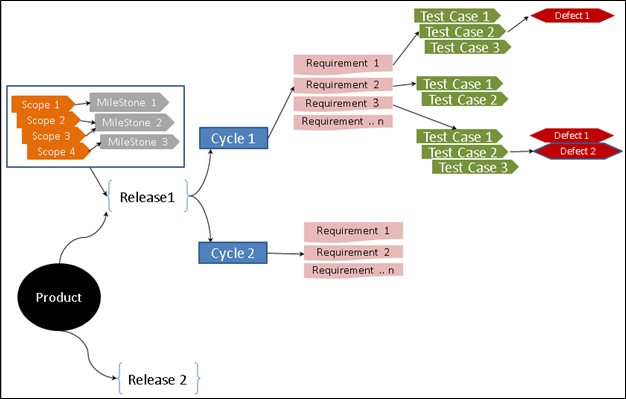
* It enables all the stakeholders to **interact and coordinate,** to achieve the project goals.
* It provides robust **tracking & reporting** and seamless integration of various project related tasks.
* It enables detailed **project analysis and effective management**.
* ALM can connect to our email systems and send emails about any changes(like Requirement change, Defect raising, etc..) to all desired team members.

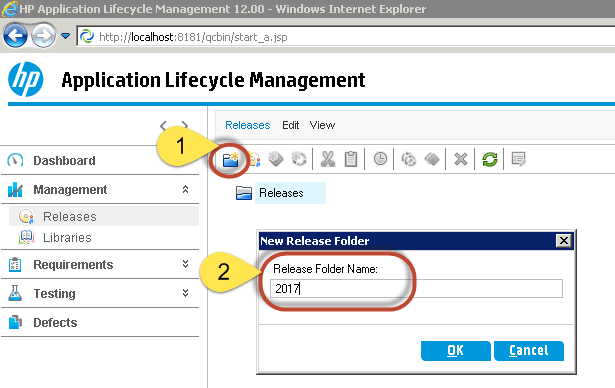
**Architecture**





* We being with planning and drafting, Release details. Determine no of Cycles in each release & Scope of each release
* For a given Release and Cycle, we draft the Requirements Specifications.
* Based on the requirements, Test plans and test cases are created.
* Next stage is executing the created tests plan
* Next stage in this test processes is tracking and fixing the defects detected in the execution stage
* During all stages, analysis is done, and reports and graphs are generated for test metric generation.





Should have- Window OS, IE browser, Oracle / MS SQL database and also MS Office suit (import excel into ALM)

**Ques - Project management tool JIRA explain?**

It is used for **bug tracking, issue tracking,** and **project management**.

Project management tool- let’s take an example of one team where people are sitting at different location, so with the help of this tool we can easily co-ordinate and communicate.

in this one number of people associated, process associated.

We can design and manage workflow also manage-project, sprints, stories created, process, defects.

Track and manage defects or issues (better reporting and better analysis).

One good thing is searching and filtering facility (we can compare last release and current release- ANAYSIS).

Also will get Dashboard view- like how many cases completed, how much work pending and all.

Create story (in backlog and move into Sprints)

Create sprint also can add watchers.

We can create subtask-because one person cannot complete, can comments.

1st close the subtask and then close the main task/ticket

**Ques - What type of testing you performed in your project?**

I performed System testing, functional testing, regression testing, smoke, sanity and UAT testing of different different products on different different channels on different different environments.

I tested commercial insurance products and for that one we performed Pricing testing, UW rule testing and Documents testing.

Commercial Insurance includes cover for almost all entities which are not categorized under Personal Insurance like Life Insurance, Health Insurance among others. Products are like

* Minifleet
* Property Owners
* Shops
* Hotel and Guest House
* Pub and Restaurant

Insurance covers are provided by the many Brands which it owns and also by partner banks few of which are mentioned in this document.

**National Insurance Group (NIG)**

**Direct Line For Business (DL4B)**

**Churchill for Business (C4B)**

**NatWest**

Each of the Brands has different Channels through which the various insurance products can be accessed directly by customers or by brokers for obtaining a cover.

**Stages of an Insurance policy –**

**Request for Quote –Quote –New Business (NB)/Policy –Mid Term Adjustment (MTA) –Cancellation –Renewals** .

**Product Testing –**

Product Testing extensively includes –

* Web UI Testing
* Underwriter Rules Testing
* Documentation Testing
* Pricing

**Web User Interface Testing –** UI Testing involves testing of the application’s build for each product in every channel. We test whether the risk questionnaire for a product, appearing on the application screens is as per the respective **‘Web Specification’** document provided by the business team. Whether all the static and dynamic questions appear correctly, headings and certain messages and wordings are as per the spec. Also, we can test the look and feel of the application.

**Underwriter (UW) Rules Testing –** Underwriting is a term used in Insurance to describe the process of assessing and evaluating the risk to decide how much coverage a client should receive, amount to be paid by the client for the cover he/she seeks and also decide whether to even accept the risk requested by a client.

For this assessment, Insurance Underwriter sets down rules and testing team tests if a product complies with the rules listed for it. Testing team uses the **UW Test Pack (TP)** prepared by the Insurance Underwriter for each product and shared to the team.

In Underwriter Rules Testing there is –

**Testing of Referral Rules –**

**Testing of Declines –**

**Testing of Endorsements –**

**Risk Update-**

**Documentation Testing -**

**Pricing Testing –**

Pricing is a test of premium amounts calculated and generated for an insurance cover.

The Pricing team and the Insurance Underwriter discuss and decide upon the rates and other necessary criteria for calculating the premium amounts for a cover requested by a customer.

Only in the quote obtained stage, when the premium matches, we can perform any overrides mentioned in the TP. Override is a certain percentage/amount of **Discount** or **Load** provided by the Insurer.

**Ques - What is testing and explain their types?**

**Ques - Black box vs white box testing?**

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. 7This type of testing is based entirely on the software requirements and specifications.

In BlackBox Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.



The above Black-Box can be any software system you want to test. For example: an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

## Types of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

* **Functional testing** - This black box testing type is related to functional requirements of a system; it is done by software testers.
* **Non-functional testing** - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
* **Regression testing** - [Regression Testing](https://www.guru99.com/regression-testing.html) is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

## Black box testing strategy:

Following are the prominent[Test Strategy](https://www.guru99.com/how-to-create-test-strategy-document.html)amongst the many used in Black box Testing

* **Equivalence Class Testing:** It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.
* **Boundary Value Testing:** Boundary value testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is mostly suitable for the systems where input is within certain ranges.
* **Decision Table Testing**: A decision table puts causes and their effects in a matrix. There is unique combination in each column.

**White Box Testing** is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.

## **What do you verify in White Box Testing?**

White box testing involves the testing of the software code for the following:

* Internal security holes
* Broken or poorly structured paths in the coding processes
* The flow of specific inputs through the code
* Expected output
* The functionality of conditional loops
* Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

## **How do you perform White Box Testing?**

To give you a simplified explanation of white box testing, we have divided it into **two basic steps**. This is what testers do when testing an application using the white box testing technique:

**STEP 1) UNDERSTAND THE SOURCE CODE**

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

**Step 2) CREATE TEST CASES AND EXECUTE**

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code. The tester will develop little tests for each process or series of processes in the application. This method requires that the tester must have intimate knowledge of the code and is often done by the developer.

**Statement Coverage** - This technique requires every possible statement in the code to be tested at least once during the testing process.

**Branch Coverage -**This technique checks every possible path (if-else and other conditional loops) of a software application

## **Types of White Box Testing**

White box testingencompasses several testing types used to evaluate the usability of an application, block of code or specific software package. There are listed below --

[**Unit Testing**](https://www.guru99.com/unit-testing-guide.html)**:**It is often the first type of testing done on an application. Unit Testing is performed on each unit or block of code as it is developed.

Unit testing is essentially done by the programmer. As a software developer, you develop a few lines of code, a single function or an object and test it to make sure it works before continuing

Unit testing helps identify majority of bugs, early in the software development lifecycle. Bugs identified in this stage are cheaper and easy to fix.

**Testing for Memory Leaks**: Memory leaks are leading causes of slower running applications. A QA specialist who is experienced at detecting memory leaks is essential in cases where you have a slow running software application.

There are many tools available to assist developers/testers with memory leak testing, example, [Rational Purify](http://www-01.ibm.com/software/awdtools/purify/win/) for windows application

## Advantages of White Box Testing

* Code optimization by finding hidden errors.
* White box tests cases can be easily automated.
* Testing is more thorough as all code paths are usually covered.
* Testing can start early in SDLC even if GUI is not available.

## Disadvantages of White Box Testing

* White box testing can be quite complex and expensive.
* Developers who usually execute white box test cases detest it. The white box testing by developers is not detailed can lead to production errors.
* White box testing requires professional resources, with a detailed understanding of programming and implementation.
* White-box testing is time-consuming, bigger programming applications take the time to test fully.