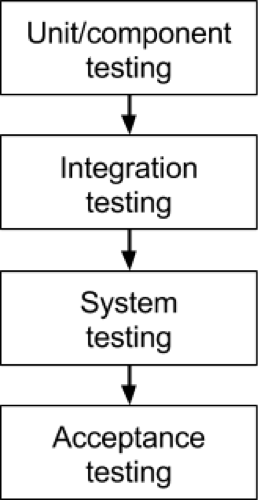
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Waterfall** | **Iterative** | **spiral** | **V model** | **Bing Bang** |
| The waterfall is a cascade SDLC model that presents the development process like the flow, moving step by step through the phases of analysis, projecting, realization, testing, implantation, and support.  Waterfall implies strict documentation.  Simple to use and understand. | The iterative model is quite different. It implies that the whole process is divided into a particular number iterations, and during each of them, developers build a limited number of features.  The progress is easy measurable. | Spiral model is a combination of the Iteartive and Waterfall SDLC models with a significant accent on the risk analysis. The main issue of the spiral model is defining the right moment to take a step into the next stage.  The customer isn’t sure about the requirements. | The V-model is called this way because of the scheme’s appearance and because its primary priorities are Verification and Validation.  Verification and validation mean different things, though they seem pretty similar.  Verification:  Product Right.  Validation:  Right Product. | The Big Bang Model is popular in situations or adopted when the project's stakeholders or customers are not certain regarding their wants and requirements.  The is quite a simple model.  Resources necessary for developing the product is only a few. |

LEVEL OF TESTING

1. Unit/Component Testing
2. Integration testing
3. System testing
4. Acceptance testing



**1. Unit/component testing**

The most basic type of testing is unit, or component, testing.

Unit testing aims to verify each part of the software by isolating it and then perform tests to demonstrate that each individual component is correct in terms of fulfilling requirements and the desired functionality.

This type of testing is performed at the earliest stages of the development process, and in many cases it is executed by the developers themselves before handing the software over to the testing team.

### 2. Integration testing

Integration testing aims to test different parts of the system in combination in order to assess if they work correctly together. By testing the units in groups, any faults in the way they interact together can be identified.

There are many ways to test how different components of the system function at their interface; testers can adopt either a bottom-up or a top-down integration method.

### 3. System testing

The next level of testing is system testing. As the name implies, all the components of the software are tested as a whole in order to ensure that the overall product meets the requirements specified.

[System testing](https://reqtest.com/testing-blog/software-system-testing/) is a very important step as the software is almost ready to ship and it can be tested in an environment which is very close to that which the user will experience once it is deployed.

### 4. Acceptance testing

Finally, [acceptance testing](https://reqtest.com/testing-blog/a-guide-to-excellent-acceptance-testing/) is the level in the software testing process where a product is given the green light or not. The aim of this type of testing is to evaluate whether the system complies with the end-user requirements and if it is ready for deployment.

The testing team will utilise a variety of methods, such as pre-written scenarios and test cases to test the software and use the results obtained from these tools to find ways in which the system can be improved.

**Principles of Testing**

1)How many principles testing?

Ans:- **There are seven principles in software testing:** 

1. Testing shows the presence of defects - shows the error or mistakes.
2. Exhaustive testing is not possible
3. Early testing
4. Defect clustering - Grouping or combining the similar types of elements.
5. Pesticide paradox - repeating the test case when we not able to find the bug.
6. Testing is context-dependent
7. Absence of errors fallacy - we are not able to say that its 100% bug free. Testing

## **Black Box Testing**

**Black box** testing is a type of software testing in which the functionality of the software is not known. The testing is done without the internal knowledge of the products.

A tester doing black-box testing has no knowledge of the software system's internal workings. Black box testing is a kind of advanced testing that focuses on the software's behavior. It entails testing from the outside or from the point of view of the end-user. Black box testing may be used at all levels of software testing, including unit, integration, system, and acceptability testing.

**Advantages of Black Box Testing:**

* The tester does not need to have more functional knowledge or programming skills to implement the Black Box Testing.
* It is efficient for implementing the tests in the larger system.
* Tests are executed from the user’s or client’s point of view.
* Test cases are easily reproducible.
* It is used in finding the ambiguity and contradictions in the functional specifications.

**Disadvantages of Black Box Testing:**

* There is a possibility of repeating the same tests while implementing the testing process.
* Without clear functional specifications, test cases are difficult to implement.
* It is difficult to execute the test cases because of complex inputs at different stages of testing.
* Sometimes, the reason for the test failure cannot be detected.
* Some programs in the application are not tested.
* It does not reveal the errors in the control structure.
* Working with a large sample space of inputs can be exhaustive and consumes a lot of time.

## Types of Black Box Testing

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

* **Functional testing** – This black box testing type is related to the 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 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.

## Tools used for Black Box Testing:

Tools used for Black box testing largely depends on the type of black box testing you are doing.

* For Functional/ Regression Tests you can use – [QTP](https://www.guru99.com/quick-test-professional-qtp-tutorial.html), [Selenium](https://www.guru99.com/selenium-tutorial.html)
* For Non-Functional Tests, you can use – [LoadRunner](https://www.guru99.com/loadrunner-v12-tutorials.html), [Jmeter](https://www.guru99.com/jmeter-tutorials.html" \t "_blank)

## **White Box Testing**

White-box testing is a kind of testing that examines the system's internal workings. Testing is based on the coverage of code statements, branches, pathways, or conditions in this approach. Low-level testing is referred to as white-box testing. Glass box, transparent box, clear box, and codebase testing are all terms used to describe this kind of testing. The white-box testing approach presumes that a unit's or program's logic route is understood.

## 7 Different types of white-box testing

1. Unit Testing
2. Static Analysis
3. Dynamic Analysis
4. Statement Coverage
5. Branch testing Coverage
6. Security Testing
7. Mutation Testing

### Unit Testing

Unit Testing is one of the basic steps, which is performed in the early stages. Most of the testers prefer performing to check if a specific unit of code is functional or not. Unit Testing is one of the common steps performed for every activity because it helps in removing basic and simple errors.

### Static Analysis

As the term says, the step involves testing some of the static elements in the code. The step is conducted to figure out any of the possible defects or errors in the application code.

### Dynamic Analysis

Dynamic Analysis is the further step of static analysis in general path testing. Most of the people prefer performing both static and dynamic at the same time.

### Statement Coverage

Statement coverage is one of the pivotal steps involved in the testing process. It offers a whole lot of advantages in terms of execution from time to time.

### Branch Testing Coverage

The modern-day software and web applications are not coded in a continuous mode because of various reasons. It is necessary to branch out at some point in time because it helps in segregating effectively.

### Security Testing

Security testing is more like a process because it comes with a lot of internal steps to complete. It verifies and rectifies any kind of unauthorized access to the system. The process helps in avoiding any kind of breach because of hacking or cracking practices.

### Mutation Testing

The last step in the process and requires a lot of time to complete effectively. Mutation testing is generally conducted to re-check any kind of bugs in the system.

## Top 10 White box Testing Tools in the market

1. [Veracode](https://www.veracode.com/)
2. [EclEmma](https://www.eclemma.org/)
3. RcUnit
4. CFIX
5. Googletest
6. EMMA
7. NUnit
8. CppUnit
9. JUnit
10. JsUnit

**Advantages of White Box testing**

1. Testing can commence even before the GUI is ready.
2. It identifies the specific procedure accuracy within the application.
3. It minutely verifies whether the program can be successfully executed with other parts of the application.
4. It identifies error in the hidden code and thus makes debugging process swift.
5. It removes extra lines of code which are not required in the program thereby optimizing the program and increases the efficiency.
6. As the internal coding of the application is considered while preparing test cases, it becomes very easy to identify the input and the expected output data.
7. It helps in evaluating all the loops and paths.
8. It can provide stability and usability of the test cases.

**Disadvantages of White Box testing**

1. As the internal code of the application has to be considered while preparing the test cases, skilled testers are required who have knowledge of programming also. Hence the cost of the resources is high.
2. It is not possible for the tester to look into every bit of the code and identify the hidden errors. This may result in failure of the application.
3. Sometimes a change in the code may be required and thus all the scenarios may need to be tested again.
4. White box testing is an exhaustive method.
5. It takes time to tester to develop the test cases.
6. Test cases are a waste if changes in the implementation code are done frequently.
7. If the application is large then complete testing through white box techniques is not feasible.

**Statement Coverage:**

**In this the  test case  is executed in such a way that every statement  of the code  is executed at least once.**

Prints (int a, int b) { ------------ Printsum is a function

int result = a+ b;

If (result> 0)

Print ("Positive", result)

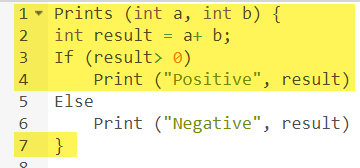
Else

Print ("Negative", result)

} ----------- End of the source code

**Scenario 1:**

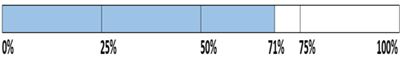
If A = 3, B = 9



The statements marked in yellow color are those which are executed as per the scenario

Number of executed statements = 5, Total number of statements = 7

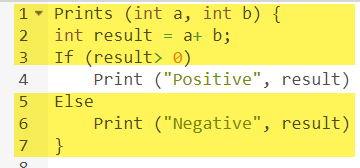
Statement Coverage: 5/7 = 71%



Likewise we will see scenario 2,

**Scenario 2:**

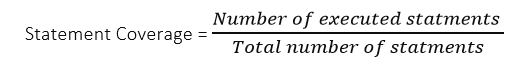
If A = -3, B = -9



The statements marked in yellow color are those which are executed as per the scenario.

Number of executed statements = 6

Total number of statements = 7



Statement Coverage: 6/7 = 85%



But overall if you see, all the statements are being covered by both scenarios. So we can conclude that overall statement coverage is 100%.



**Branch/Decision Coverage:**

**Test coverage criteria requires enough test cases such that each condition in a decision takes on all possible outcomes at least once, and each point of entry to a program or subroutine is invoked at least once.  That is, every branch (decision) taken each way, true and false. It helps in validating all the branches in the code making sure that no branch leads to abnormal behavior of the application.**

a decision point has two decision values one is true, and another is false that's why most of the times the total number of outcomes is two. The percent of decision coverage can be found by dividing the number of exercised outcome with the total number of outcomes and multiplied by 100.

Decision Coverage technique in whitebox testing link

Difference between Black Box Testing and White Box Testing

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | |  | |  |
| 1. | BBT  It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it. |  | WBT  It is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software. |
| 2. | Implementation of code is not needed for black box testing. |  | Code implementation is necessary for white box testing. |
| 3. | It is mostly done by software testers. |  | It is mostly done by software developers. |
| 4. | No knowledge of implementation is needed. |  | Knowledge of implementation is required. |
| 5. | It can be referred to as outer or external software testing. |  | It is the inner or the internal software testing. |
| 6. | It is a functional test of the software. |  | It is a structural test of the software. |
| 7. | This testing can be initiated based on the requirement specifications document. |  | This type of testing of software is started after a detail design document. |

|  |
| --- |
|  |

Difference Between Verification and Validation

|  |  |
| --- | --- |
|  |  |
| Verification is the static testing. | Validation is the dynamic testing. |
| It does *not* include the execution of the code. | It includes the execution of the code. |
| Methods used in verification are reviews, walkthroughs, inspections and desk-checking. | Methods used in validation are Black Box Testing, White Box Testing and non-functional testing. |
| It checks whether the software conforms to specifications or not. | It checks whether the software meets the requirements and expectations of a customer or not. |
| It can find the bugs in the early stage of the development. | It can only find the bugs that could not be found by the verification process. |
| The goal of verification is application and software architecture and specification. | The goal of validation is an actual product. |
| Quality assurance team does verification. | Validation is executed on software code with the help of testing team. |
| It comes before validation. | It comes after verification. |
| It consists of checking of documents/files and is performed by human. | It consists of execution of program and is performed by computer. |

**Gray Box Testing**

Gray box testing (a.k.a grey box testing) is a method you can use to debug software and evaluate vulnerabilities. In this method, the tester has limited knowledge of the workings of the component being tested. This is in contrast to [black box testing](https://www.imperva.com/learn/application-security/black-box-testing/), where the tester has no internal knowledge, and [white box testing](https://www.imperva.com/learn/application-security/white-box-testing/), where the tester has full internal knowledge.

### Matrix Testing

Matrix testing is a technique that examines all variables in an application. In this technique, technical and business risks are defined by the developers and a list of all application variables are provided. Each variable is then assessed according to the risks it presents. You can use this technique to identify unused or un-optimized variables.

### Pattern Testing

Pattern testing is a technique that evaluates past defects to identify patterns that lead to defects. Ideally, these evaluations can highlight which details contributed to defects, how the defects were found, and how effective fixes were. You can then apply this information to identifying and preventing similar defects in new versions of an application or new applications with similar structures.

### Orthogonal Array Testing

Orthogonal array testing is a technique you can use when your application has only a few inputs that are too complex or large for extensive testing. This technique enables you to perform test case optimization, where the quality and number of tests performed balance test coverage with effort. This technique is systematic and uses statistics to test pair-based interactions.

**Tools For Grey Box Testing**

**These are enlisted below:**

1. Selenium
2. Appium
3. Postman
4. JUnit
5. NUnit
6. Cucumber
7. DBUnit
8. Burp Suite
9. RestAssured

**Steps for Gray Box Testing**

The steps of Gray Box Testing are:

* Firstly the inputs are identified.
* After identifying inputs, the outputs are identified.
* Thereafter identify the major paths.
* After the major paths are identified, figure out the subfunctions.
* Once all the subfunctions are figured out, develop inputs for them.
* After developing inputs, develop the outputs.
* When the inputs and outputs are developed, execute the test cases.
* Verify the results.
* Repeat the steps for accuracy.

### Advantages and Disadvantages of Gray Box Testing

#### Advantages

* Gray box testing gives the advantages of black-box testing and white box testing.
* The product quality is better because the inputs of developers and testers are considered.
* The testing process is comparatively shorter.
* It provides the developer to fix defects.
* The testing goals are clear which makes it an easier process for developers as well as testers.
* Testers are not required to be experts.
* Gray box testing is cheaper than integration testing.
* Testing is done on the basis of high-level database diagrams and data flow diagrams.

#### Disadvantages

* Gray box testing cannot be used for the testing of algorithms.
* It might be difficult to design the test cases.
* Even though it is a combination of black box and [**white box testing**](https://openxcell.com/blog/white-box-testing/), it does not have the full benefits of white box testing.
* There might be difficulty in understanding the link between a defect and its root cause.

|  |  |  |
| --- | --- | --- |
| **Black Box Testing** | **White Box Testing** | **Grey Box Testing** |
| Internal structure of the code is not known. | Internal structure of the code is known. | Internal structure of the code is partly known. |
| Also known as closed box testing, data driven testing, functional testing. | Also known as clear box testing, structural testing, closed testing. | Also known as translucent testing. |
| Algorithm testing cannot be  done. | Algorithm testing can be done and is recommended. | Algorithm testing cannot be done. |
| Hidden errors are difficult to detect. | Hidden errors are easily detected. | Difficult to discover hidden errors. |
| The testing is done by the tester, developer and user. | The testing is done by the tester and developer. | The testing is done by the tester, developer and user. |
| Least time consuming process. | Most time consuming process. | Takes less time than black box. |
| The least exhaustive type of testing. | Most exhaustive form of testing. | It is partially exhaustive. |
| It has the largest testing space of inputs. | It has the smallest testing space of inputs. | The testing space is larger than white box testing but smaller than white box testing. |

Test Cases:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr  No. | Test case  id | **Purpose** | **Pre-Req** | **Test Data** | **Steps** | **Status** |
| 1 | TC\_001 | To give access to the application. | Users should have a user name and password to access the app. | User name: Correct Format for User name  Password: Correct password  Submit: Correct user name and password | 1. Open Application 2. Visit Login Page(Register  if Required) 3. Enter user name 4. Enter password 5. Tap on login Button 6. See the Main Page | Pass |
| 2 | TC\_002 | To give access to Choose a restaurant from the list. | Users should have to enter a user name and password to access login and then choose the restaurant. | Choose a restaurant from the navigation drawer | 1. Tap on Navigation Drawer 2. Tap on Restaurants 3. Choose your Favorite Restaurant | Pass |
| 3 | TC\_003 | To give access to order your favorite meal. | User can choose a meal from the menu list to enter its quantity and then place order | Choose your meal from menu list and place order | 1. Tap on the navigation drawer 2. Tap on Restaurants 3. Choose your Favorite Restaurant 4. Open restaurant 5. Select menu 6. Choose category 7. Select item 8. Add it to cart 9. Open orders 10. Place order 11. Pay cash after Delivery | Pass |
| 4 | TC\_004 | To give access to Add your favorite meal to favorites. | User can choose a meal from menu list add it into favorites | Choose your meal from the menu list and add it to favorites | 1. Tap on the navigation drawer 2. Tap on Restaurants 3. Choose your Favorite Restaurant 4. Open restaurant 5. Select menu 6. Choose category 7. Select item 8. Add it to favorites | Pass |
| 5 | TC\_005 | To give access to Delete meals from favorites. | Users can choose a meal from favorites and delete it. | Choose a meal from favorites and delete it | 1. Tap on the navigation drawer 2. Tap on Restaurants 3. Choose your Favorite Restaurant 4. Open restaurant 5. Select favorites 6. Tap on dish 7. Alert Dialogue box open 8. Two options DELETE or CANCEL 9. Tap on Delete Button | Pass |
| 6 | TC\_006 | To give access to Direct contact with Restaurant | User can choose a contact button after choosing retaurant. | Contact with restaurant | 1. Tap on navigation drawer 2. Tap on Restaurants 3. Choose your Favorite Restaurant 4. Open restaurant 5. Select contact 6. Tap on Call or Message | Pass |
| 7 | TC\_007 | To give access to Manage the restaurant. | Manager can enter user name and password to access its app. | User name: Correct user name  Password : Correct password  Submit button: Correct user name and password to access. | 1. Open admin application 2. Enter user name 3. Enter password 4. Click on submit button 5. See full menu 6. Manage tables 7. Manage orders |  |

what is a test case?

A test case is a document which consists of a set of conditions or actions which are performed on the software application in order to verify the expected functionality of the feature. Here we describe the end to end logical flow of a specific requirement with test data, prerequisites and expected results.

Typical Test Case Parameters:

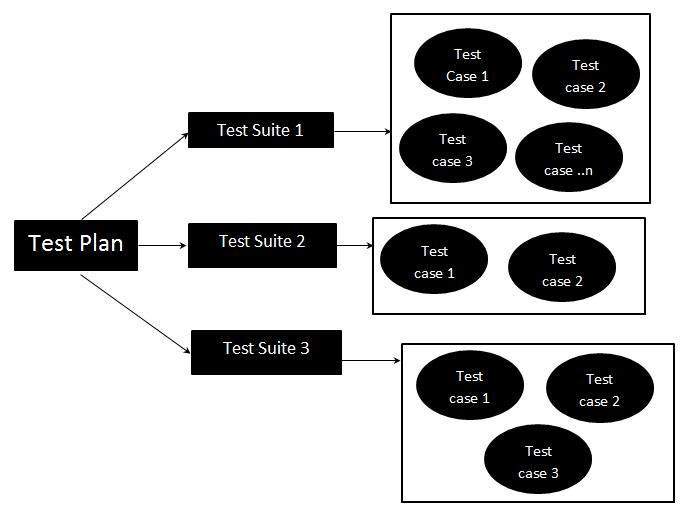
## What is a Test Scenario?

A **Test Scenario** is defined as any functionality that can be tested. It is also called *Test Condition* or *Test Possibility*. As a tester, you should put yourself in the end user’s shoes and figure out the real-world scenarios and use cases of the Application Under Test.

## **What is a Test Suite?**

Test suite is a container that has a set of tests which helps testers in executing and reporting the test execution status. It can take any of the three states namely Active, Inprogress and completed.

## **Test Suite - Diagram:**



What is Test Basis?

Test basis is defined as **the source of information or the document that is needed to write test cases and also for test analysis**. Test basis should be well defined and adequately structured so that one can easily identify test conditions from which test cases can be derived.

Performance Testing:

Performance Testing is a software testing process used for testing the speed, response time, stability, reliability, scalability, and resource usage of a software application under a particular workload. The main purpose of performance testing is to identify and eliminate the performance bottlenecks in the software application. It is a subset of performance engineering and is also known as “Perf Testing”.

The focus of Performance Testing is checking a software program’s:

1.Speed

2.Scalability

3.Stability

4.Relablity

1.Speed: Determines whether the application responds quickly

2.Scalability: Determines the maximum user load the software application can handle.

3. Stability: Determines if the application is stable under varying loads.

4.Realbility: Reliability Testing is an important software testing technique that is performed by the team to ensure that the software is performing and functioning consistently in each environmental condition as well as in a specified period. It ensures that product is fault free and is reliable for its intended purpose.

## Types of Performance Testing

* **Load testing –** checks the application’s ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.
* **Stress testing –** involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.
* **Endurance testing –** is done to make sure the software can handle the expected load over a long period of time.
* **Spike testing –** tests the software’s reaction to sudden large spikes in the load generated by users.
* **Volume testing** – Under Volume Testing large no. of. Data is populated in a database, and the overall software system’s behavior is monitored. The objective is to check software application’s performance under varying database volumes.
* **Scalability testing**– The objective of scalability testing is to determine the software application’s effectiveness in “scaling up” to support an increase in user load. It helps plan capacity addition to your software system.



Performance Testing workflow

### Step 1) Identify Your Testing Environment

Know your physical test environment, production environment and what testing tools are available. Understand details of the hardware, software and network configurations used during testing before you begin the testing process

### Step 2) Identify the Performance Acceptance Criteria

This includes goals and constraints for throughput, response times and resource allocation. It is also necessary to identify project success criteria outside of these goals and constraints.

### Step 3) Plan & Design Performance Tests

Determine how usage is likely to vary amongst end users and identify key scenarios to test for all possible use cases. It is necessary to simulate a variety of end users, plan performance test data and outline what metrics will be gathered.

### Step 4) Configuring the Test Environment

Prepare the testing environment before execution. Also, arrange tools and other resources.

### Step 5) Implement Test Design

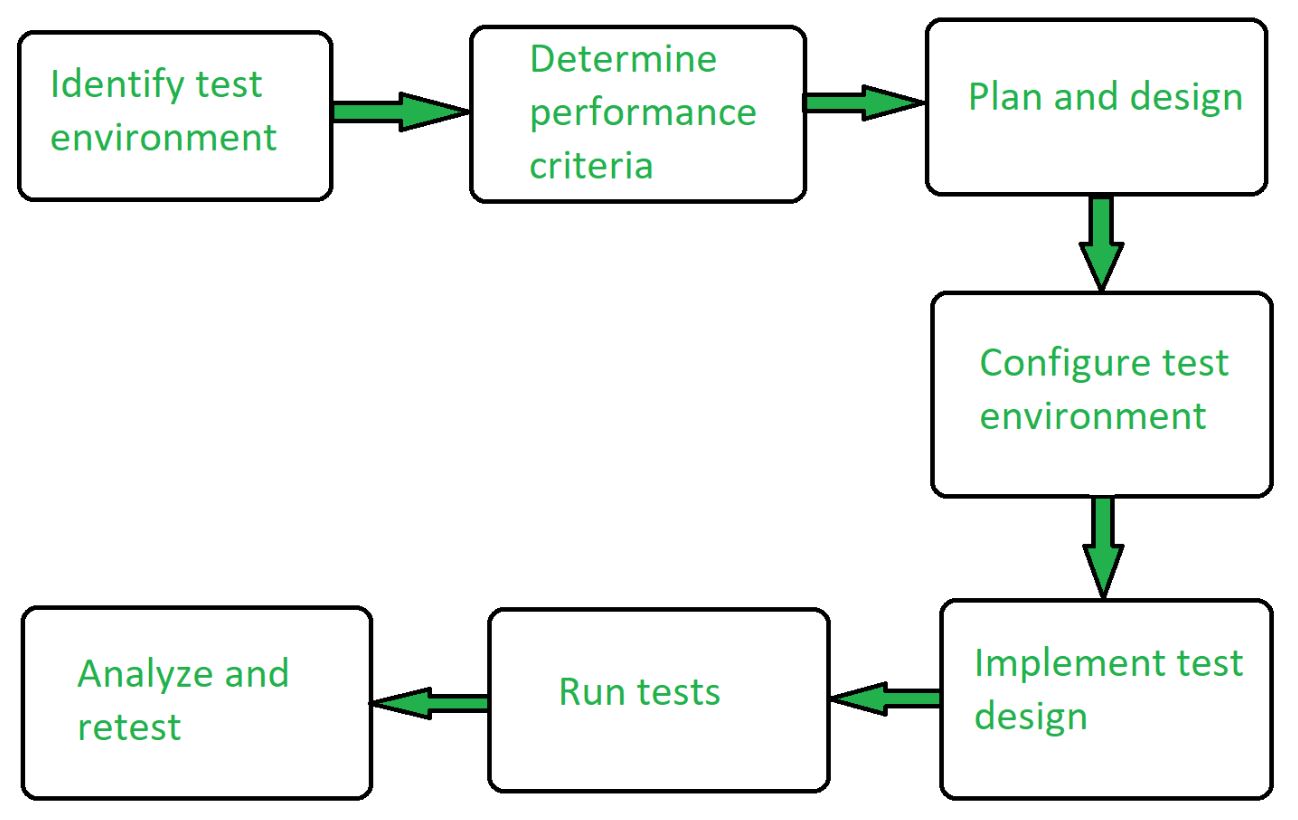
Create the performance tests according to your test design.

### Step 6) Run the Tests

Execute and monitor the tests.

### Step 7) Analyze, Tune and Retest

Consolidate, analyze and share test results. Then fine tune and test again to see if there is an improvement or decrease in performance.



**FUNCTIONAL TESTINGS:**

1.Assertion testing

2.Gorilla testing

3.Sanity testing

4.Monkey testing

5.Smoke testing

6.Exploratory testing

7.Mutation testing

8.Benchmark testing

**1.ASSERTION TESTING:**

**Definition:**

An **assertion** is a **boolean expression**.it is a concept of **functional testing**. It is used to test a **logical expression**. An assertion is true if the logical expression that is being tested is true and there are no bugs in the program. Assertion testing can be used at any particular stage of the program.

EX: 3<4=true, assertion true and logical expression true then their is no bugs.

7>10=false, assertion false and logical expression false then bugs are their.

**Benefits of Assertions:**

The main advantage of having assertions is to identify defects in a program. The usefulness of assertions include:

* It is used to detect subtle errors which might go unnoticed.
* It is used to detect errors sooner after they occur.
* Make a statement about the effects of the code that is guaranteed to be true.

**Limitations:**

* Failing to report a bug that exists.
* Reporting an error when it does not exist.
* Can lead to other side effects
* Can Take time to execute if it contains errors and occupies memory as well.

**2.GORILLA TESTING:**

Gorilla testing is a software testing technique that repeatedly applies inputs on a module to ensure it is functioning correctly and that there are no bugs.

**3.SANITY TESTING:**

Sanity testing is performed on stable builds and it is also known as a variant of regression testing.

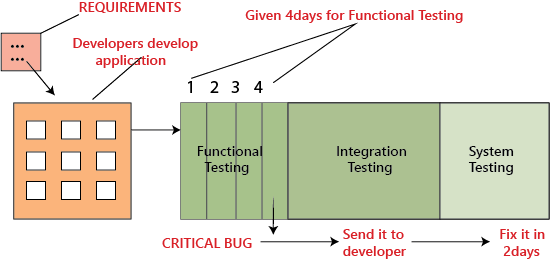
Sanity testing was performed when we are receiving software build (with minor code changes) from the development team. It is a checkpoint to assess if testing for the build can proceed or not.

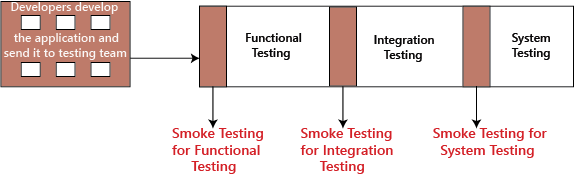
**4.MONKEY TSETING:**

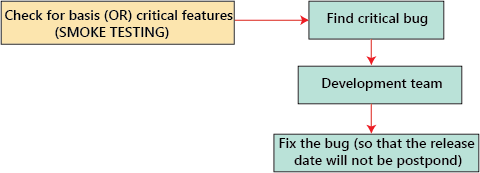
Monkey Testing is a software testing technique in which the tester enters any random inputs into the software application without predefined test cases and checks the behavior of the software application, whether it crashes or not. The purpose of Monkey testing is to find the bugs and errors in the software application using experimental techniques.

**5.SMOKE TESTING:**

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as “Build Verification Testing” or “Confidence Testing.”







**6.EXPLORATOY TESTING:**

exploratory testing?

If requirement does not exist, then we do one round of exploratory testing.

So, for this first, we will be exploring the application in all possible ways, understanding the flow of the application, preparing a test document and then testing the application, this approach is known as exploratory testing.

**7.MUTATION TESTING:**

Mutation Testing is a type of software testing in which certain statements of the source code are changed/mutated to check if the test cases are able to find errors in source code. The goal of Mutation Testing is ensuring the quality of test cases in terms of robustness that it should fail the mutated source code.

**8. Fuzz Testing** or Fuzzing is a software testing technique of putting invalid or random data called FUZZ into software system to discover coding errors and security loopholes. The purpose of fuzz testing is inserting data using automated or semi-automated techniques and testing the system for various exceptions like system crashing or failure of built-in code, etc.

**# NON-FUNCTIONAL TESTINGS:**

1.Load testing

2.Strees testing

3.Usability testing

4.Performance testing

5.Volume testing

6.Scalability testing

7.Security testing

**1.LOAD TSETING:**

Load Testing is a non-functional software testing process in which the performance of software application is tested under a specific expected load. It determines how the software application behaves while being accessed by multiple users simultaneously. The goal of Load Testing is to improve performance bottlenecks and to ensure stability and smooth functioning of software application before deployment.

**Load Testing Tools:**

1. Apache JMeter

2. Web Load

3. Neo Load

4. Load Ninja

5. HP Performance Tester

6. Load UI Pro

7. Load View

**2.STRESS TESTING:**

Stress testing (sometimes called torture testing) is a form of deliberately intense or thorough testing used to determine the stability of a given system, critical infrastructure or entity. Stress testing involves testing the application under varying load. Extremely large numbers of concurrent users try to log into the application. Database linked to the website shuts down when the website tries to reach it from the front end. Data in added in extremely large quantity in the database. Stress Testing is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn’t crash under crunch situations.

**3.USABILITY TSETING:**

Usability testing refers to evaluating a product or service by testing it with representative users. Typically, during a test, participants will try to complete typical tasks while observers watch, listen and takes notes.

to check the usability or ease of using a software product. Checking the user-friendliness, efficiency, and accuracy of the application is known as **Usability Testing.**

**Parameters:**

Efficiency

Memorability

Accuracy

Learnability

Satisfaction

Errors

**4.PERFORMANCE TESTING:**

Performance testing is in general a testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

Types of Performance Testing:

• Load

• Stress

• Spike

• Endurance

• Scalability

• Volume

**5.VOLUME TESTING:**

Volume Testing is a type of software testing which is carried out to test a software application with a certain amount of data.

In volume testing a huge volume of data is acted upon the software. It is basically performed to analyze the performance of the system by increasing the volume of data in the database.

Volume Testing is also known as Flood Testing.

**6.SCALABILITY TESTING:**

scalability testing, which comes under the non-functional testing of software testing.

It is used to check an application's performance by increasing or decreasing the load in particular scales known as scalability testing. It is executed at a hardware, software, or database level.

**7.SECURITY TESTING:**

The main goal of Security Testing is to identify the threats in the system and measure its potential vulnerabilities, so the threats can be encountered and the system does not stop functioning or can not be exploited

types of security Vulnerability Scanning. ...

Security Scanning. ...

Penetration Testing. ...

Security Audit/ Review. ...

Ethical Hacking. ...

Risk Assessment. ...

Posture Assessment. ...

Authentication.

**8. BENCHMARK TESTING:**

A Benchmark in Performance Testing is a metric or a point of reference against which software products or services can be compared to assess the quality measures

Benchmark Testing is not a term related to just software testing, but it also deals with Hardware Testing

**Soak Testing** is a type of software testing in which system is tested under huge load over a continuous availability period to check the behavior of the system under production use.   
Soak Testing tests that system can withstand a huge volume of the load for an extended period of time.

**Objective of Soak Testing:**   
The objective of Soak Testing is to:

* To check the system behavior under heavy load for long time.
* To predict the failure caused by the heavy load.
* To test the performance of the system.
* To make the system reliable and stable.

**Failures detected by Soak Testing:**   
The failures or issues detected by the Soak Testing are:

* **Memory Leaks:**   
  Soak testing detects the serious memory leaks which can cause application crash or lead up to the crash of operating system.
* **Layer Connections Failure:**   
  Soak testing finds the failure of close connections between the layers of the system that can interrupt the modules of the system.
* **Database Connections Failure:**   
  Soak testing detects the failure of close database connections under some conditions that may crash the complete system.
* **Response Time Degradation:**   
  Soak testing finds the degradation of response time of the system as the system becomes less efficient and takes more time to response.

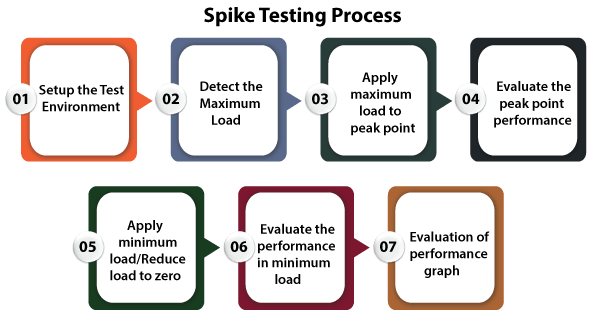
**Advantages of Soak Testing:** The advantages of Soak Testing are:

* Soak Testing improves the performance of the system.
* Soak Testing increases the resistance of the system.
* Soak Testing makes the system work under heavy load.
* It improves the behavior of the system under the heavy load for long time.

**Disadvantages of Soak Testing:**

* It is difficult to know or predict how long that the test will run.
* Utilization of the memory is high due to more number of users accessing the web application.
* It is a time consumption process and it is not recommended for the project which has strict deadlines.
* Manual soak testing often takes lot of time to complete the test and often provides wrong test results.
* If we run this technique in a live environment then it will lead to loss of data or data corruption.

**Spike testing** is **a type of performance testing in which an application receives a sudden and extreme increase or decrease in load**. The goal of spike testing is to determine the behavior of a software application when it receives extreme variations in traffic.



**Connection testing:**

An internet speed test **measures the connection speed and quality of your connected device to the internet**. It does so by running multiple consecutive tests that analyze different aspects of your internet connection, namely ping (latency), download speed, and upload speed.

**Production testing**

Testing in production, rather, refers to **the continuous testing of the application in the production environment, after a deployment**.

J meter

* Ability to load and [performance test](https://www.optisolbusiness.com/services/performance-testing-services) many different applications/server/protocol types
* Full-featured Test IDE that allows fast Test Plan recording
* CLI mode to load test from any Java-compatible OS
* A complete and ready to present dynamic HTML report
* Easy correlation through the ability to extract data from most popular response formats, HTML, JSON, XML or any textual format
* Complete portability and 100% Java purity
* Full multithreading framework
* Caching and offline analysis/replaying of test results

**Security Testing**

**Security Testing** is a type of [Software Testing](https://www.geeksforgeeks.org/software-testing-basics/) that uncovers vulnerabilities of the system and determines that the data and resources of the system are protected from possible intruders. It ensures that the software system and application are free from any threats or risks that can cause a loss. Security testing of any system is focused on finding all possible loopholes and weaknesses of the system which might result in the loss of information or repute of the organization.

**Goal of Security Testing:** The goal of security testing is to:

* To identify the threats in the system.
* To measure the potential vulnerabilities of the system.
* To help in detecting every possible security risks in the system.
* To help developers in fixing the security problems through coding.

**Principle of Security Testing:** Below are the six basic principles of security testing:

* Confidentiality
* Integrity
* Authentication
* Authorization
* Availability
* Non-repudiation

**Major Focus Areas in Security Testing:**

* Network Security
* System Software Security
* Client-side Application Security
* Server-side Application Security

#### **Types Of security Testing**

#### **Vulnerability Scanning**

Often powered by automation (manual tools exist too), [**vulnerability scanning**](https://www.indusface.com/blog/online-vulnerability-scanner/) is leveraged to identify known loopholes and vulnerability signatures. It is the first of many steps in vulnerability management and app/ software security. It is used to gain an understanding of the baseline of security risks.

#### **Security Scanning**

[**Security scanning**](https://www.indusface.com/web-application-scanning.php/) is the process of identifying vulnerabilities and misconfigurations in the app/ software, network, and systems. Both manual and automated tools are used for this test type. The insights from these tests are listed, analyzed in-depth, and solutions provided to fix the issue.

#### **Penetration Testing**

[**Penetration Testing**](https://www.indusface.com/learning/what-is-penetration-testing/) (Pen-Testing) is the process of stimulating a real-time cyberattack against an app/ software, system or network under secure conditions. It is (and must be) performed manually by a trusted, certified security expert to understand the strength of the security measures against attacks in real-time. Most importantly, unknown vulnerabilities (including zero-day threats and business logic flaws) are exposed through Pen-Testing.

#### **Security Audit/ Review**

Security auditing or security review is the structured process to review/ audit the app/software against defined standards. Through gap analysis and code/ design reviews, the security of the physical configurations, operating system, information handling processes, user practices, etc. is assessed. Compliance with regulatory standards and frameworks is assessed as well.

#### **Ethical Hacking**

[**Ethical hacking**](https://www.indusface.com/blog/ethical-hacking/), broader than penetration testing, is an umbrella term that includes a multitude of hacking methodologies. Here, all vulnerabilities and misconfigurations are attempted to be exposed by simulating attacks from within the app/ software.

#### **Risk Assessment**

Through risk assessments, the security risks facing the app/ software/ network are identified, analyzed, and classified (as Critical, High, Medium, Low). Mitigation measures and controls are recommended thereon, based on the priority.

#### **Posture Assessment**

The overall security posture of the organization is assessed through posture assessment using a combination of security scanning, ethical hacking, and risk assessment.

.

**The 7 Attributes That Security Testing Must Include:**

#### **Authentication**

The user is digitally identified before getting access to the system through authentication. By testing and validating this attribute, the system’s efficacy in allowing only legitimate/ right users is ensured. The system could use a straightforward Username-Password or Multi-Factor authentication process (where a combination of OTP, biometrics, secure ID tokens, etc. could be used).

#### **Authorization**

Once the user is authenticated, they gain access to the system. Their privileges and permissions to perform actions within the system is defined based on user roles and limited by authorization. For instance, it is determined by the authorization attribute if a specific user can modify data, access certain files, and so on.

#### **Confidentiality**

By testing the confidentiality attribute, it is verified whether the information, services, and resources are accessible only to intended users and only when requested. The tester can

* identify if unauthorized users are accessing privileged resources.
* verify if all data is encrypted.
* analyze the format in which data is displayed when requested, etc.

#### **Availability**

When the availability attribute is tested, the tester can understand if the software/ app is up and running round-the-clock with minimal accepted downtimes (from regular maintenance and upgrades). The availability of information and services upon request and backup files in case of failures is verified too.

#### **Integrity**

It is verified through the integrity attribute if

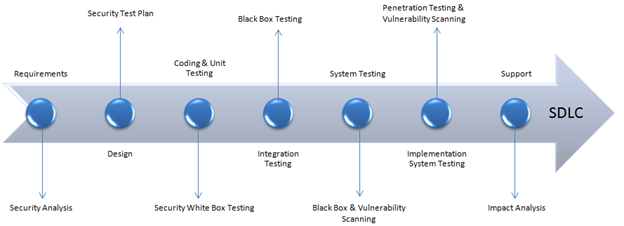
* information received is unaltered in transit.
* correct and updated information is presented as per user groups, privileges, and restrictions.

#### **Non-Repudiation**

Here, the denied access requests along with Timestamp and IP address are tracked. It is confirmed by the tester if the user is genuine and not a [**security threat**](https://www.indusface.com/blog/top-threats-a-web-application-firewall-can-mitigate/).

#### **Resilience**

The resistance to face internal and external attacks by the system is checked through the testing of the resilience attribute.



## Example Test Scenarios for Security Testing:

Sample Test scenarios to give you a glimpse of security test cases –

* A password should be in encrypted format
* Application or System should not allow invalid users
* Check cookies and session time for application
* For financial sites, the Browser back button should not work.

In security testing, different methodologies are followed, and they are as follows:

* **Tiger Box**: This hacking is usually done on a laptop which has a collection of OSs and hacking tools. This testing helps penetration testers and security testers to conduct vulnerabilities assessment and attacks.
* [**Black Box**](https://www.guru99.com/black-box-testing.html): Tester is authorized to do testing on everything about the network topology and the technology.
* **Grey Box**: Partial information is given to the tester about the system, and it is a hybrid of white and black box models.

## Security Testing Roles

* Hackers – Access computer system or network without authorization
* Crackers – Break into the systems to steal or destroy data
* Ethical Hacker – Performs most of the breaking activities but with permission from the owner
* Script Kiddies or packet monkeys – Inexperienced Hackers with programming language skill

## Security Testing Tool

### 1) [Acunetix](https://bit.ly/3qH5T77" \t "_blank)

Intuitive and easy to use, [Acunetix](https://bit.ly/3qH5T77" \t "_blank) by Invicti helps small to medium-sized organizations ensure their web applications are secure from costly data breaches. It does so by detecting a wide range of web security issues and helping security and development professionals act fast to resolve them.

**Features:**

* Advanced scanning for 7,000+ web vulnerabilities, including OWASP Top 10 such as SQLi and XSS
* Automated web asset discovery for identifying abandoned or forgotten websites
* Advanced crawler for the most complex web applications, incl. multi-form and password-protected areas
* Combined interactive and dynamic application security testing to discover vulnerabilities other tools miss
* Proof of exploit provided for many types of vulnerabilities
* DevOps automation through integrations with popular issue tracking and CI/CD tools
* Compliance reporting for regulatory standards, such as PCI DSS, NIST, HIPAA, ISO 27001, and more.

### 2) [Intruder](https://guru99.live/qxoGpg)

[Intruder](https://guru99.live/qxoGpg) is a powerful, automated penetration testing tool that discovers security weaknesses across your IT environment. Offering industry-leading security checks, continuous monitoring and an easy-to-use platform, Intruder keeps businesses of all sizes safe from hackers.

**Features:**

* Best-in-class threat coverage with over **10,000 security checks**
* Checks for configuration weaknesses, missing patches, application weaknesses (such as SQL injection & cross-site scripting) and more
* Automatic analysis and prioritisation of scan results
* Intuitive interface, quick to set-up and run your first scans
* Proactive security monitoring for the latest vulnerabilities
* AWS, Azure and Google Cloud connectors
* API integration with your CI/CD pipeline

### 3) Owasp

The Open Web Application Security Project ([OWASP](https://owasp.org/www-project-web-testing-environment/)) is a worldwide non-profit organization focused on improving the security of software. The project has multiple tools to pen test various software environments and protocols. Flagship tools of the project include

### 4) WireShark

[Wireshark](https://bit.ly/2TMN561) is a network analysis tool previously known as Ethereal. It captures packet in real time and display them in human readable format. Basically, it is a network packet analyzer- which provides the minute details about your network protocols, decryption, packet information, etc. It is an open source and can be used on Linux, Windows, OS X, Solaris, NetBSD, FreeBSD and many other systems. The information that is retrieved via this tool can be viewed through a GUI or the TTY mode TShark Utility.

### 5) W3af

[w3af](https://bit.ly/2P5Qrm7) is a web application attack and audit framework. It has three types of plugins; discovery, audit and attack that communicate with each other for any vulnerabilities in site, for example a discovery plugin in w3af looks for different url’s to test for vulnerabilities and forward it to the audit plugin which then uses these URL’s to search for vulnerabilities.



Vulnerability

## Vulnerability Testing

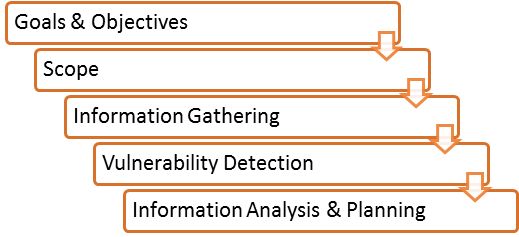
**Vulnerability Testing** also called Vulnerability Assessment is a process of evaluating security risks in software systems to reduce the probability of threats. The purpose of vulnerability testing is reducing the possibility for intruders/hackers to get unauthorized access of systems. It depends on the mechanism named Vulnerability Assessment and Penetration Testing (VAPT) or VAPT testing.

A vulnerability is any mistake or weakness in the system’s security procedures, design, implementation or any internal control that may result in the violation of the system’s security.

* It is important for the security of the organization.
* The process of locating and reporting the vulnerabilities, which provide a way to detect and resolve security problems by ranking the vulnerabilities before someone or something can exploit them.
* In this process Operating systems, Application Software and Network are scanned in order to identify the occurrence of vulnerabilities, which include inappropriate software design, insecure authentication, etc.

## Vulnerability Assessment Process

Here is the step by step **Vulnerability Assessment Process** to identify the system vulnerabilities.



**Step 1) Goals & Objectives** : – Define goals and objectives of Vulnerability Analysis.

**Step 2) Scope** : – While performing the Assessment and Test, Scope of the Assignment needs to be clearly defined.

The following are the three possible scopes that exist:

* [Black Box Testing](https://www.guru99.com/black-box-testing.html) : – Testing from an external network with no prior knowledge of the internal network and systems.
* Grey Box Testing : – Testing from either external or internal networks with the knowledge of the internal network and system. It’s the combination of both Black Box Testing and White Box Testing.
* [White Box Testing](https://www.guru99.com/white-box-testing.html) : – Testing within the internal network with the knowledge of the internal network and system. Also known as Internal Testing.

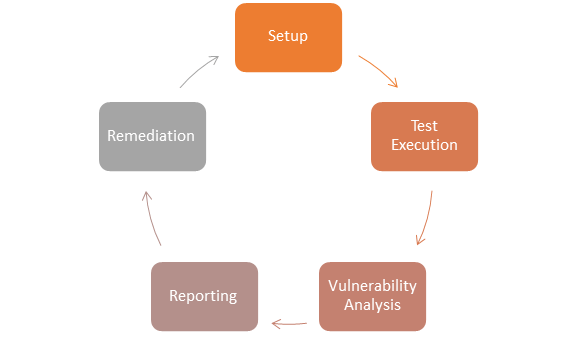
**Step 3) Information Gathering** : – Obtaining as much information about IT environment such as Networks, IP Address, Operating System Version, etc. It’s applicable to all the three types of Scopes such as Black Box Testing, Grey Box Testing and White Box Testing.

**Step 4) Vulnerability Detection** : – In this process, vulnerability scanners are used to scan the IT environment and identify the vulnerabilities.

**Step 5) Information Analysis and Planning** : – It will analyze the identified vulnerabilities to devise a plan for penetrating into the network and systems.

## How to do Vulnerability Assessment

Following is the step by step process on **How to do Vulnerability Assessment**:



**Step 1) Setup:**

* Begin Documentation
* Secure Permissions
* Update Tools
* Configure Tools

**Step 2) Test Execution:**

* Run the Tools
* Run the captured data packet (A packet is the unit of data that is routed between an origin and the destination. When any file, for example, e-mail message, HTML file, Uniform Resource Locator(URL) request, etc. is sent from one place to another on the internet, the TCP layer of TCP/IP divides the file into a number of “chunks” for efficient routing, and each of these chunks will be uniquely numbered and will include the Internet address of the destination. These chunks are called packets. When all the packets are arrived, they will be reassembled into the original file by the TCP layer at the receiving end while running the assessment tools

**Step 3) Vulnerability Analysis:**

* Defining and classifying network or System resources.
* Assigning priority to the resources( Ex: – High, Medium, Low)
* Identifying potential threats to each resource.
* Developing a strategy to deal with the most prioritized problems first.
* Defining and implementing ways to minimize the consequences if an attack occurs.

**Step 5) Remediation:**

* The process of fixing the vulnerabilities.
* Performed for every vulnerability

## Types of a vulnerability scanner

1. **Host Based**

* Identifies the issues in the host or the system.
* The process is carried out by using host-based scanners and diagnose the vulnerabilities.
* The host-based tools will load a mediator software onto the target system; it will trace the event and report it to the security analyst.

1. **Network-Based**

* It will detect the open port, and identify the unknown services running on these ports. Then it will disclose possible vulnerabilities associated with these services.
* This process is done by using Network-based Scanners.

1. **Database-Based**

* It will identify the security exposure in the database systems using tools and techniques to prevent from SQL Injections. (SQL Injections: – Injecting SQL statements into the database by the malicious users, which can read the sensitive data’s from a database and can update the data in the Database.)

## Tools for Vulnerability Scanning

### 1) [Intruder](https://guru99.live/1z8JAX)

[Intruder](https://guru99.live/1z8JAX) is a powerful vulnerability assessment tool that discovers security weaknesses across your IT environment. Offering industry-leading security checks, continuous monitoring and an easy-to-use platform, Intruder keeps businesses of all sizes safe from hackers.

**Features:**

* Best-in-class threat coverage with over **10,000 security checks**
* Checks for configuration weaknesses, missing patches, application weaknesses (such as SQL injection & cross-site scripting) and more
* Automatic analysis and prioritisation of scan results
* Intuitive interface, quick to set-up and run your first scans
* Proactive security monitoring for the latest vulnerabilities
* AWS, Azure and Google Cloud connectors
* API integration with your CI/CD pipeline

### 2) [Acunetix](https://bit.ly/3jAMVh3" \t "_blank)

Intuitive and easy to use, [Acunetix](https://bit.ly/3jAMVh3" \t "_blank) by Invicti helps small to medium-sized organizations ensure their web applications are secure from costly data breaches. It does so by detecting a wide range of web security issues and helping security and development professionals act fast to resolve them.

**Features:**

* Advanced scanning for 7,000+ web vulnerabilities, including OWASP Top 10 such as SQLi and XSS
* Automated web asset discovery for identifying abandoned or forgotten websites
* Advanced crawler for the most complex web applications, incl. multi-form and password-protected areas
* Combined interactive and dynamic application security testing to discover vulnerabilities other tools miss
* Proof of exploit provided for many types of vulnerabilities
* DevOps automation through integrations with popular issue tracking and CI/CD tools
* Compliance reporting for regulatory standards, such as PCI DSS, NIST, HIPAA, ISO 27001, and more.

## Advantages of Vulnerability Assessment

* Open Source tools are available.
* Identifies almost all vulnerabilities
* Automated for Scanning.
* Easy to run on a regular basis.

## Disadvantages of Vulnerability Assessment

* High false positive rate
* Can easily detect by Intrusion Detection System Firewall.
* Often fail to notice the latest vulnerabilities.

## Vulnerability Testing Methods

**Active Testing**

* Inactive Testing, a tester introduces new test data and analyzes the results.
* During the testing process, the testers create a mental model of the process, and it will grow further during the interaction with the software under test.
* While doing the test, the tester will actively involve in the process of finding out the new test cases and new ideas. That’s why it is called Active Testing.

**Passive Testing**

* Passive testing, monitoring the result of running software under test without introducing new test cases or data

**Network Testing**

* Network Testing is the process of measuring and recording the current state of network operation over a period of time.
* Testing is mainly done for predicting the network operating under load or to find out the problems created by new services.
* We need to Test the following Network Characteristics:-
* Utilization levels
* Number of Users
* Application Utilization

**Distributed Testing**

* Distributed Tests are applied for testing distributed applications, which means, the applications that are working with multiple clients simultaneously. Basically, testing a distributed application means testing its client and server parts separately, but by using a distributed testing method, we can test them all together.
* The test parts will interact with each other during the Test Run. This makes them synchronized in an appropriate manner. Synchronization is one of the most crucial points in distributed testing.

31/10/2022

* SECURITY TESTING TOOLS

As part of Security Testing, the testing team can take the help of different tools for performing:

* **Vulnerability Assessment**: As part of this the security testing team will identify the vulnerabilities in the system and provide a detailed assessment report with the list of identified vulnerabilities.
* **Penetration Testing**: As part of this the security testing team will exploit the identified vulnerabilities, to check whether they can be compromised to gain access and control on the system.

**Types of Security Testing Tools**: Security Testing tools can be categorized into two types:

* **Scanners**: These tools are used for identifying the vulnerabilities.
* **Attackers**: These tools are used for attacking the system with an intention of compromising the identified vulnerabilities to gain access and complete control.

**Security Testing Tools**: The following are the different scanning and attacking tools for performing Security Testing:

The following are some of the Security testing tools:

1. **Zed Attack Proxy (ZAP)**
2. **SonarQube**
3. **Wapiti**
4. **Netsparker**
5. **Arachni**
6. **Iron Wasp**
7. **Grabber**
8. **SQLMap**
9. **Wfuzz**
10. **W3af**

### ****1. Zed Attack Proxy (ZAP)****

ZAP, or Zed Attack Proxy, is a multi-platform, open-sourceonline application security testing tool developed by OWASP (Open Web Application Security Project). During the development and testing phases of a web app, ZAP is used to uncover a variety of security flaws.

Application error disclosure

* Cookie not HttpOnly flag
* SQL injection
* Application error disclosure
* XSS injection
* Missing anti-CSRF tokens and security headers
* Private IP disclosure
* Cookie not HttpOnly flag
* Session ID in URL rewrite

**Key Features:**

* For advanced users, it will support command-line access.
* It has the capability of being used as a scanner.
* It will perform web application scanning automatically.
* It works with a variety of operating systems, including Windows, OS X, and Linux.
* It takes advantage of AJAX spiders, which are both powerful and old.

### ****2. SonarQube****

Sonar Source created this open-source security tool. It is used to verify the quality of code and run automated reviews on web applications written in various programming languages such as [Java](https://www.geeksforgeeks.org/java/), [C#](https://www.geeksforgeeks.org/csharp-programming-language/), [JavaScript](https://www.geeksforgeeks.org/introduction-to-javascript/), [PHP](https://www.geeksforgeeks.org/php/), Ruby, Cobol, [C](https://www.geeksforgeeks.org/c-language-set-1-introduction/)/[C++](https://www.geeksforgeeks.org/c-plus-plus/), and so on by discovering bugs, code analysis, and security exposures.

* It will use SonarLint plug-ins to interface with a variety of development environments, including Visual Studio, Eclipse, and IntelliJ IDEA.
* External technologies such as GitHub, LDAP, and Active Directory are also supported.
* It can keep track of metric history and provide graphs of evolution.
* It will assist us in identifying the more complicated issues.
* It will ensure the security of the application.

### ****3. Wapiti****

Wapiti is a free, open-source project from SourceForge and develop that is one of the leading web application security testing tools. Wapiti uses black-box testing to look for security vulnerabilities in online applications. GET and POST HTTP attack methods are supported by the open-source security testing tool. Wapiti exposes the following vulnerabilities:

* Command Execution detection
* CRLF injection
* Database injection
* File disclosure
* Shellshock or Bash bug
* SSRF (Server Side Request Forgery)
* Weak .htaccess configurations that can be bypassed
* XSS injection
* XXE injection

**Key Features:**

* Allows for several types of authentication, such as Kerberos and NTLM.
* It includes a buster module that allows you to brute force directory and file names on the webserver you’re targeting.
* It works in the same way that a fuzzer would.
* Attacks can be carried out using both the GET and POST HTTP protocols.

### ****4. Netsparker****

It is used to detect the web application’s vulnerabilities in a unique way, as well as to verify whether the application’s weaknesses are correct or erroneous.Netsparker is capable of scanning any web application, regardless of the platform or programming language used to build it.

**Key Features:**

* It will scan all forms of legacy as well as new online applications such as Web 2.0, HTML5, and SPA (single page apps).
* It will provide a variety of out-of-the-box reports for both developers and management for various objectives.
* With the help of our templates, we can create unique reports.
* To safeguard our application, we can use this tool in conjunction with CI/CD platforms like Bamboo, Jenkins, or TeamCity.

### ****5. Arachni****

Arachni is a web application security scanner that is suitable for both penetration testers and administrators. This open-sourcesecurity testing program may detect a variety of flaws, including the following:

* Invalidated redirect
* Local and remote file inclusion
* SQL injection
* XSS injection

**Key Features:**

* Immediately deployable
* Ruby framework that is modular and high-performing
* Support for several platforms

### 6. Iron Wasp

Iron Wasp is a strong open-source scanning tool that can detect over 25 different types of web application flaws. It can also distinguish between false positives and false negatives. Iron Wasp aids in the discovery of a wide range of flaws, including:

* Broken authentication
* Cross-site scripting
* CSRF
* Hidden parameters
* Privilege escalation

**Key Features:**

* C#, Python, Ruby, or VB.NET are used to extend the system via plugins or modules.
* HTML and RTF formats are used to create reports.

### ****7. Grabber****

The Grabber is a simple web application scanner that can be used to search forums and personal websites. The Python-based lightweight security testing tool has no graphical user interface. Grabber discovered the following vulnerabilities:

* Backup files verification
* Cross-site scripting
* File inclusion
* Hidden parameters
* Privilege escalation
* Simple AJAX verification
* SQL injection

**Key Features:**

* Produces a statistics analysis file.
* Simple and easy to transport
* Supports the examination of JS code.

### 8. SQLMap

SQLmap is an open-source tool for detecting and exploiting SQL injection problems in penetration testing. SQLmap is a tool that automates the detection and use of SQL injection. SQL Injection attacks have the ability to gain control of SQL databases. They can harm any website or online program that uses a SQL database, including MySQL, SQL Server, Oracle, and a variety of others. The security testing tool has a robust testing engine that can support six different SQL injection techniques:

* Boolean-based blind
* Error-based
* Out-of-band
* Stacked queries
* Time-based blind
* UNION query

**Key Features**

* This tool automates the process of locating SQL injection flaws.
* It can also be used to test a website’s security.
* A powerful detecting engine
* MySQL, Oracle, and PostgreSQL are among the databases supported.

### ****9. Wfuzz****

Wfuzz is a tool for brute-forcing Web applications. It can be used to find non-linked directories, servlets, scripts, and other resources, as well as brute-force, GET and POST parameters for checking various types of injections (SQL, XSS, LDAP, and so on), brute-force Forms parameters (User/Password), and fuzzing. Vulnerabilities exposed by Wfuzz are:

* LDAP injection
* SQL injection
* XSS injection

**Key Features:**

* Numerous injection sites with multiple dictionaries, HTML output, recursion (when performing directory brute force attacks), colored outputs with formatting, and so on are some of the capabilities of this application.
* Other features include brute-forcing posts, headers, authentication data, fuzzing cookies, time delays between requests, and support for SOCK/authentication/proxy.
* Wfuzz also allows you to combine payloads with iterators, perform HEAD scans, use brute force HTTP methods (POST), use several proxy servers (each request goes through a separate proxy), and hide results using return codes, word numbers, line numbers, and responses or regex.

### 10. W3af

The open-source w3af (web application attack and audit framework) web application security scanner. The project offers a Web application vulnerability scanner and exploitation tool. It gives information about security flaws that can be used in penetration testing projects. A graphical user interface and a command-line interface are also available on the scanner.

**Key Features:**

* Support for authentication
* It’s simple to get started with and has a user-friendly interface.
* The output can be saved to a terminal, a file, or sent through email.

# What is a Staging Environment?

A staging environment or staging site is a copy of your live website and is the last step in the [deployment process](https://umbraco.com/knowledge-base/deployment/) before changes are deployed to your live website.

## Environments: Local vs. development vs. staging vs. live

### Local environment

A local environment is independent of any hosting as it is kept offline and run on a local machine or server. This is an efficient and cost-effective way of adding a test environment without worrying about paying for multiple environments. It also adds the benefit of being independent on internet connectivity, so you can work on your project wherever and whenever.

### Development environment

A [development environment](https://umbraco.com/knowledge-base/development-environment/) is usually used as an “anything-goes” area for developers to test new features, changes, or anything they feel like. This environment can sometimes be the same as the local environment, depending on the preferred setup and work process.

### Staging environment

A staging environment is the last step before something goes into production and is visible on the live site.

A staging site’s main purpose is to ensure that all new changes deployed from previous environments are working as intended before they hit the live website. By using a staging site and testing everything before deploying to a live website, you will be able to eliminate bugs and issues, so they never affect the user. Sometimes this process is referred to as quality assessment (QA).

### Live environment

The live environment - also called the production environment - is the final stage and the website that a user will see. If the process of deploying and testing changes is good between your environments, then this version of your website should be without bugs or issues, so you can provide the end-user with a great digital experience.

### Left to right deployment workflow

## **Test Environment Management Tools**

### *Apwide Golive*

Apwide Golive is a test environment management hub that works in Jira. This tool is an add-on that allows organizations to increase the visibility of their test environments and perform scheduling and orchestration of environments. It comes with a REST API that allows users to integrate it with their already existing toolchain.

**Pros:**Great integration with Jira and most CI/CD tools using its REST API.

**Cons:**Apwide Golive is a Jira add-on, so you must use Jira. Also, there is no free tier.

**Site:** [www.apwide.com](https://www.apwide.com/)

### *Omnium Lite*

Omnium Lite describes itself as a test environment management DevSecOps toolset. With the help of this tool, organizations can automate booking, scheduling, and requesting IT environments for development, production, and testing needs.

Omnium Lite offers monitoring and tracing. Users have real-time visibility into their environments and a calendar view of booked environments.

**Pros:** Automatic integrations out of the box, strong security capabilities, and real-time visibility.

**Cons:**A free trial is offered, but there’s no free version.

**Site:** [www.temscorp.com](https://www.temscorp.com/index.html)

### ServiceNow Test Management

ServiceNow Test Management is an application by ServiceNow that allows users to manage [manual testing](https://www.testim.io/blog/exploratory-testing-guide/) sessions. It offers the possibility of requesting and booking test environments with the help of Orchestration, another application by ServiceNow.

**Pros:**It offers integrations with Jira and Microsoft Azure DevOps.

**Cons:**It mainly targets manual testing.

**Site:** [www.servicenow.com](https://www.servicenow.com/)

### 08/11/2022

### SEQUENTIAL DIAGRAM

**Use-case diagram:**

Passenger Admin

#### **Basic Flow:**

**Actors**

This use case starts when an actor wishes to log into the Ticket Registration System.

1. The system requests that the Actor enter his/her name and password.
2. The actor enters his/her name and password.
3. The system validates the entered name and password and logs the actor into the system.

#### **Alternative Flows:**

**Invalid Name / Password**  
If in the *Basic Flow*the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to either return to the beginning of the *Basic Flow*or cancel the login, at which point the use case ends.

#### **Pre-Conditions**

None

#### **Post-Conditions**

If the use case was successful, the actor is now logged into the system. If not the system state is unchanged.

|  |  |  |
| --- | --- | --- |
| Use cases | Actors | Description |
| 1 | Login | Login screen is having enter username/email address with submit and reset button. |
| 2 | verification | User should not be allowed to Login with Invalid Username and Valid Password. |
| 3 | validation | Appropriate validation message should be displayed at correct place when providing invalid Username / Password. |
|  |  |  |
|  | Passengers indicates that they would like to purchase tickets. | System requests the following data from the Passenger:   * Airline * Number of tickets |
|  | Passenger fills out data. | The System verifies that the above 2 items have been filled out.  If any data is missing the system warns the passenger and the scenario continue with software reaction.  If all data has been entered, the system asks the customer to verify their purchase. |
|  |  |  |

### Online Food Ordering System - You can edit this template and create your own diagram. Creately diagrams can be exported and added to Word, PPT (powerpoint), Excel, Visio or any other document. Use PDF export for high-quality prints and SVG export for large sharp images. #UML #UseCase #UseCaseDiagram #OnlineSystem #FoodOrderingSystem #Diagrams Sequence Diagram, Class Diagram, Process Flow Chart, Food Ordering App, Restaurant Order, Diagram Design, Order Food Online, Business Analysis, Erd