**Snehal Thomake**

**Date= 07/11/2022**

**Batch=7670(Thane)**

**Test Environment Management**

## ****What is a Test Environment?****

A test environment is a set of hardware on which a software application or product under test (AUT) and supporting software are deployed to assist in testing. In [SDLC](https://www.softwaretestingmaterial.com/sdlc-software-development-life-cycle/), testing is an essential part as it ensures the quality of the product. That’s why different [software testing](https://www.softwaretestingmaterial.com/software-testing/) types are performed to check diverse parameters or test cases.

During the testing phases, many members are involved, such as developers, testers, and sometimes customers or clients. A test environment is critical to testing since it provides the perfect setup for testing teams. It concerns different testing tools and other elements. In short, it supports test execution with hardware, software, and network configured.

## ****What is a Test Bed?****

A test execution environment configured for testing is known as test bed. It consists of specific hardware, operating system, network configuration, etc. It helps to verify [test cases](https://www.softwaretestingmaterial.com/test-case-template-with-explanation/), which require a certain amount of data setup. For example, suppose you want to test whether a particular function created invoices for sales data. In that case, it’s important to [create the database](https://www.softwaretestingmaterial.com/sql-create-database/), and this test environment is considered a test bed. Here’re a few points, which tell about test bed:

* It ensures you have enough data set up before you start the testing.
* It also ensures that the team is aware of the kind of data required as well as if the data needs to be provided by other teams.

## ****Types of Software Development Environments****

1. Development environment
2. Testing environment
3. Staging environment
4. Production environment

### ****1. Development Environment****

It is the main branch of a software application. This is where developers spend most of their time in order to write the first lines of code. From here, an app transforms from concept into the MVP (Minimum Viable Product).

### ****2. Testing Environment****

It is a set-up of software and hardware for the testing teams to execute test cases. In other words, a test environment supports test execution with hardware, software, and network configured. This is also a type of testing where application testing is conducted to find and fix errors.

**Following are the key areas of setting up the testing environment:**

* Test Data
* Database Server
* Browser
* System & applications
* Network
* Hardware that includes Server Operating System

### ****3. Staging Environment****

This type of testing is where all the work is done in the development environment and merged into the built system. It is often used to automate the process of software compilation before it is moved into the production environment.

### ****4. Production Environment****

The last type of environment in software development is the production environment, where new updates or builds are moved into production for end users.

## ****Key Elements for Creating a Test Environment****

A test environment consists of the following elements:

* Configure the environment
* Set up database
* Test Data
* Network configuration
* Opt for the right hardware and operating system, e.g., knowing the difference between running an app on different versions of operating systems.

Moreover, one of the most important elements for creating a test environment is documenting all the actions. This is the key for other users to be able to copy the environment. Also, the detailed documentation enables testers to set up different test environments like staging and production environments.

### Types

As it turns out, there are quite a lot of them. Which one to use depends on the size and needs of each organization. Common categorizations of test environments include the following:

* **Dev environment.**The development environment is where engineers write the code to implement features and fix bugs. Hopefully, it’s also where they write [unit tests](https://www.testim.io/blog/unit-testing-best-practices/) and, maybe, [integration tests](https://www.testim.io/blog/unit-test-vs-integration-test/). Typically, this “environment” consists of the developers’ machines. Often, in cloud or containers/microservices environments, the dev playground is a smaller and similar version of the production environment.
* **Test or QA environment.** This could refer to an environment where the app is deployed for testing after it has passed initial testing checkpoints. You might have testers and QA analysts performing and writing different kinds of tests in the QA environment.
* **Pre-production.**This is an environment where the product owner or a customer representative can perform acceptance testing to “sign off on” features for release. The whole testing process might start in this environment with some acceptance or [BDD-style tests](https://www.testim.io/blog/tdd-vs-bdd-a-developers-pocket-reference-with-examples/) before the features are implemented. It should be a close approximation of the production environment.
* **Production.** The production environment is also a hotbed of testing. Sometimes production is used for acceptance or exploratory testing. Production can also be used in phased roll-outs using feature flagging to expose features to a select group of users before expanding the deployment to the full set of users.

## 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.

### 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.

### Service Now Test Management

Service Now Test Management is an application by Service Now 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 Service Now.

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

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