

Configuration Testing Tutorial: Sample Test Cases & Example

What is Configuration Testing?

Configuration testing is the method of testing an application with multiple combinations of software and hardware to find out the optimal configurations that the system can work without any flaws or bugs.

As discussed above, Configuration Testing is a software testing where the application under test has to be tested using multiple combinations of Software and Hardware.

Configuration Testing Example

Let's understand this with an example of a Desktop Application:

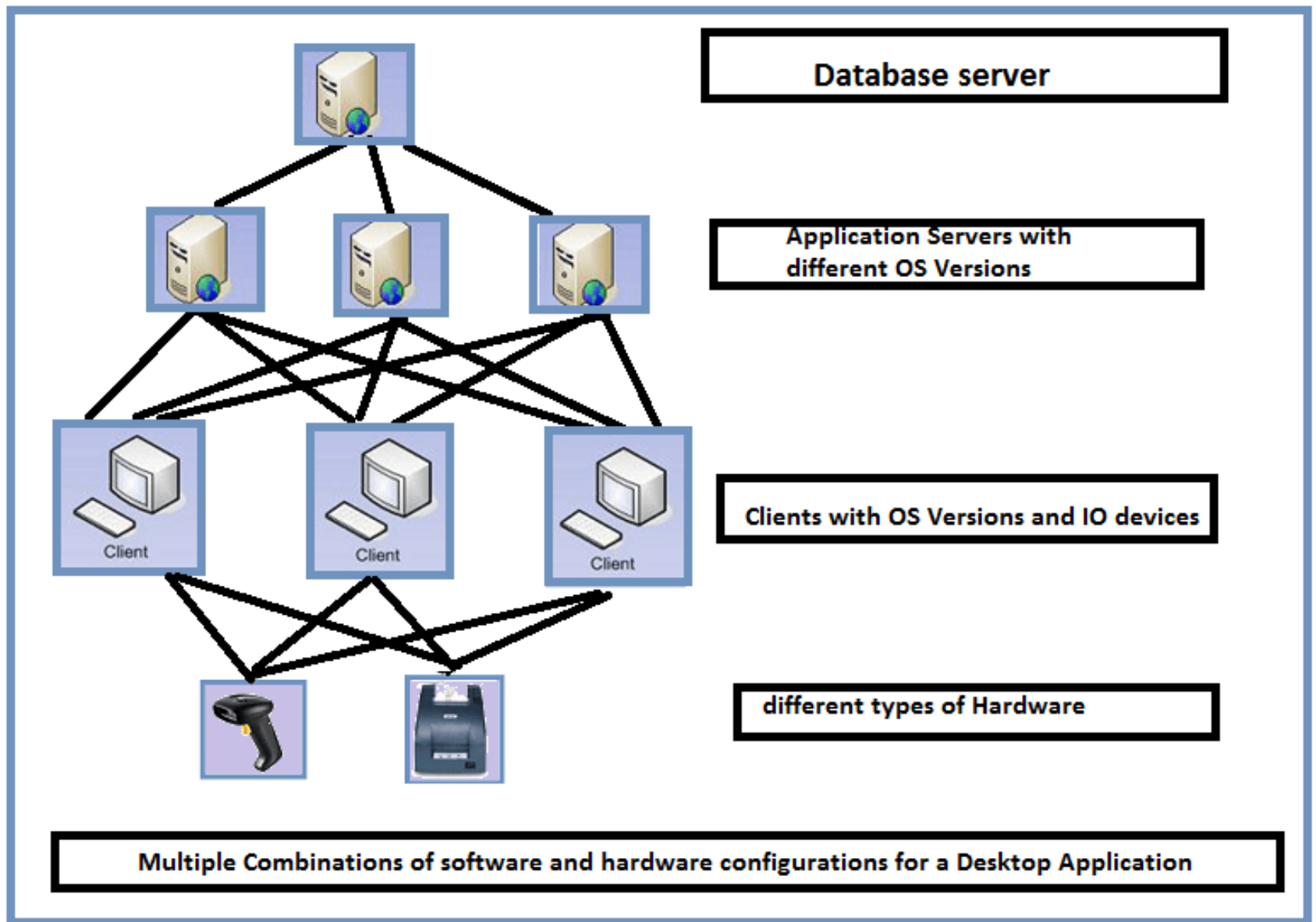
Generally, Desktop applications will be of 2 tier or 3 tier, here we will consider a 3 tier Desktop application which is developed using [Asp.Net](#) and consists of Client, Business Logic Server and Database Server where each component supports below-mentioned platforms.

- Client Platform - Windows XP, Window7 OS, windows 8 OS , etc
- Server Platform - Windows Server 2008 R2, Windows Server 2008 R2, Windows Server 2012R2
- Database –SQL Sever 2008, [SQL](#) Server 2008R2, SQL Server 2012, etc.

A tester has to test the Combination of Client, Server and Database with combinations of the above-mentioned platforms and database versions to ensure that the application is functioning properly and does not fail.

Configuration testing is not only restricted to Software but also applicable for Hardware which is why it is also referred as a Hardware configuration testing, where we test different hardware devices like Printers, Scanners,

Web cams, etc. that support the application under test.



Pre-requisites for Configuration Testing

For any project before starting with the configuration testing, we have to follow some pre-requisites

- Creation of matrix which consists of various combinations of software and hardware configurations
- Prioritizing the configurations as its difficult to test all the configurations
- Testing every configuration based on prioritization.

Objectives of Configuration Testing

The objectives of configuration Testing is to

- Validating the application to determine if it fulfills the configurability requirements

- Manually causing failures which help in identifying the defects that are not efficiently found during testing (Ex: changing the regional settings of the system like Time Zone, Language, Date time formats, etc.)
- Determine an optimal configuration of the application under test.
- Analyzing the system performance by adding or modifying the hardware resources like Load Balancers, increase or decrease in memory size, connecting various printer models, etc.
- Analyzing system Efficiency based on the prioritization, how efficiently the tests were performed with the resources available to achieve the optimal system configuration.
- Verification of the system in a geographically distributed Environment to verify how effectively the system performs.

For Ex: Server at a different location and clients at a different location, the system should work fine irrespective of the system settings.

- Verifying how easily the bugs are reproducible irrespective of the configuration changes.
- Ensuring how traceable the application items are by properly documenting and maintaining the versions which are easily identifiable.
- Verifying how manageable the application items are throughout the software development life cycle.

Configuration Testing - Test Strategy

In this section, we will discuss the strategy that needs to be followed for configuration testing types and there are two types of configuration testing as mentioned below

- Software Configuration Testing
- Hardware Configuration Testing

Software Configuration Testing

Software configuration testing is testing the Application under test with multiple OS, different software updates, etc. Software Configuration testing is very time consuming as it takes time to install and uninstall different software's that is used for the testing.

One of the approaches that is followed to test the software configuration is to test on Virtual Machines. Virtual Machine is an Environment that is installed on software and acts like a Physical Hardware and users will have the same feel as of a Physical Machine. Virtual Machines simulates real-time configurations.

Instead of Installing and uninstalling the software in multiple physical machines which is time-consuming, it's always better to install the application/software in the virtual machine and continue testing. This process can be performed by having multiple virtual machines, which simplifies the job of a tester

Software configuration testing can typically begin when

- Configurability requirements to be tested are specified
- Test Environment is ready
- Testing Team is well trained in configuration testing
- Build released is unit and Integration test passed

Typical [Test Strategy](#) that is followed to test the software configuration test is to run the functional test suite across multiple software configurations to verify if the application under test is working as desired without any flaws or errors.

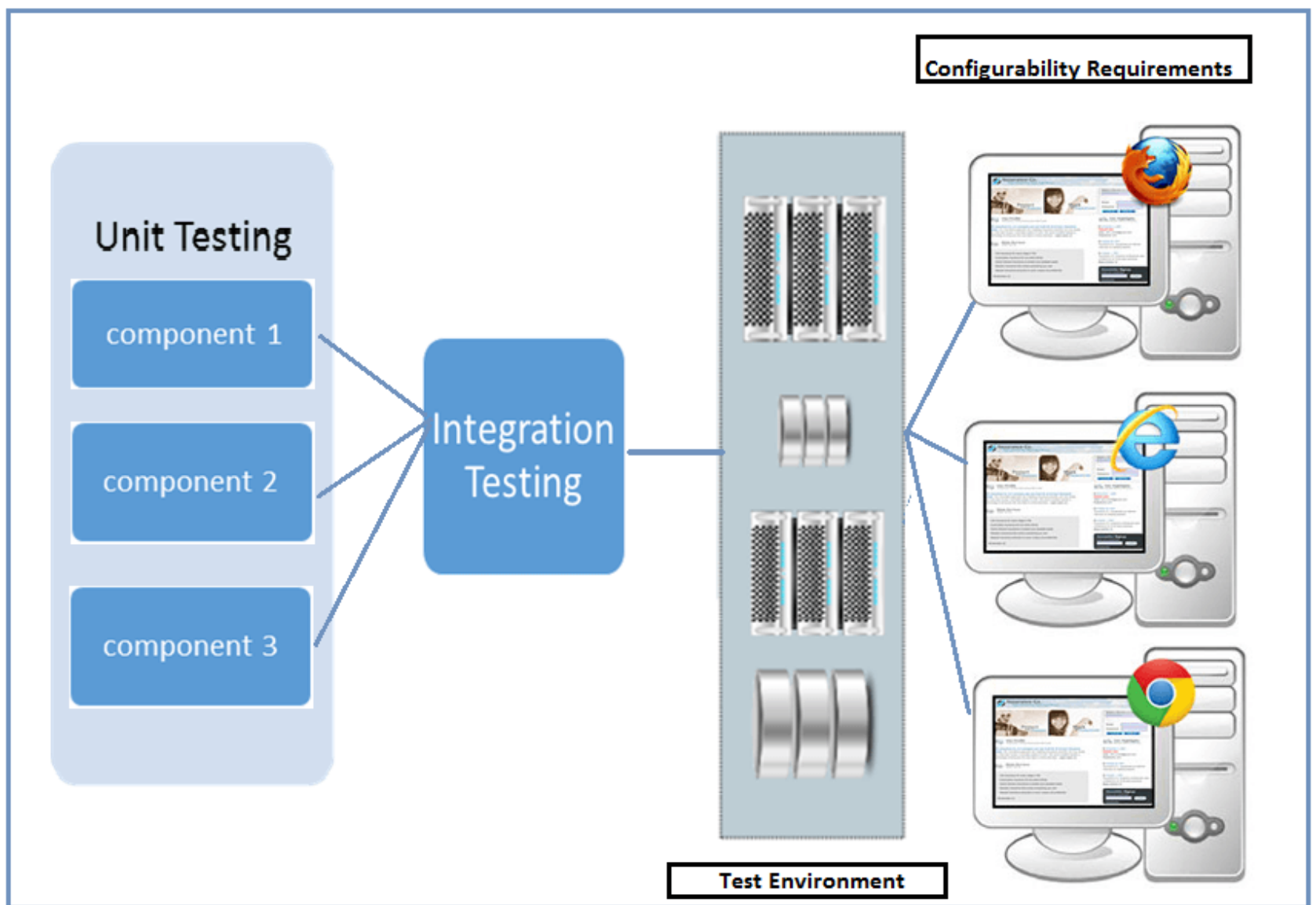
Another strategy is to ensure the system is working fine by manually failing the test cases and verifying for the efficiency.

Example:

Say there is a Banking Application, which has to be tested for its compatibility across multiple browsers when the application is hosted in an environment where all the prerequisites are present it might pass the

unit and [Integration Testing](#) in the test lab.

But if the same application is installed in a client place and the machines are missing some software's updates or the versions on which the application is dependent directly or indirectly there is a chance that the application might fail. To avoid this kind of situation, it's always suggested to fail the tests manually by removing some of the configurability requirements and then proceed with the testing.



Hardware Configuration Testing

Hardware configuration testing is generally performed in labs, where we find physical machines with different hardware attached to them.

Whenever a build is released, the software has to be installed in all the physical machines where the hardware is attached, and the test suite has to be run on each machine to ensure that the application is working fine.

To perform the above task a significant amount of effort is required to install the software on each machine, attach the hardware and manually

running or even to automate the above said process and running the test suite.

Also, while performing hardware configuration test, we specify the type of hardware to be tested, and there are a lot of computer hardware and peripherals which make it quite impossible to run all of them. So it becomes the duty of the tester to analyze what hardware is mostly used by users and try to make the testing based on the prioritization.

Sample Test Cases

Consider a Banking Scenario to test for the hardware compatibility. A Banking Application that is connected to Note Counting Machine has to be tested with different models like Rolex, Strob, Maxsell, StoK, etc.

Let's take some sample test cases to test the Note Counting Machine

- Verifying the connection of application with Rolex model when the prerequisites are NOT installed
- Verifying the connection of application with Rolex model when the prerequisites are installed
- Verify if the system is counting the notes correctly
- Verify if the system is counting the notes correctly
- Verifying the tampered notes
- Verifying the response times
- Verifying if the fake notes are detected and so on

The above test cases are for one model, and the same has to be tested with all the models available in the market by setting them up in a test lab which is difficult. Hence, it is advisable to outsource the hardware configuration testing to organizations which specializes them.

Summary:

Configuration Testing should be given with equal importance like other testing types. Without configuration testing being performed it is difficult to analyze the optimal system performance and also software might

encounter compatibility issues that it is supposed to run on.