**Types of performance testing:**

* Load Testing
* Stress Testing
* Endurance Testing
* Spike Testing
* Volume Testing
* Scalability Testing

**Load Testing:**

As the best known and most commonly conducted type of performance testing, load testing involves applying ordinary stress to a software application or IT system to see if it can perform as intended under normal conditions. It is related to its bigger, more brutal cousin, stress testing, but load testing ensures that a given function, program, or system can simply handle what it’s designed to handle, whereas stress testing is about overloading things until they break, applying unrealistic or unlikely load scenarios. Both practices can play important roles in determining exactly how well a given piece of frontend software, such as a website, or a backend system, such as the Apache server hosting that site, can deal with the actual loads they’re likely to encounter through regular use. Stress testing deliberately induces failures so that you can analyze the risk involved at the breaking points, and then, perhaps, choose to tweak programs to make them break more gracefully. Stress testing is useful for preparing for the unexpected and determining exactly how far a given system can be pushed, exploring the outer limits of performance capacity. But when it comes to simply making sure that a software application or physical network can endure the user requests and actions it is likely to encounter in ordinary circumstances, load testing is the right method for the task.

Of course, it should be noted that if your application isn’t actually ready for the expected demands, then a test that was intended to be a load test when you launched it can suddenly become a stress test while it’s running. Once the load starts causing things to break, from that moment on you are, by definition, stressing the system. This is the main reason the terms are often confused, because the exact same test may turn out to be a load test under some situations and a stress test under others.

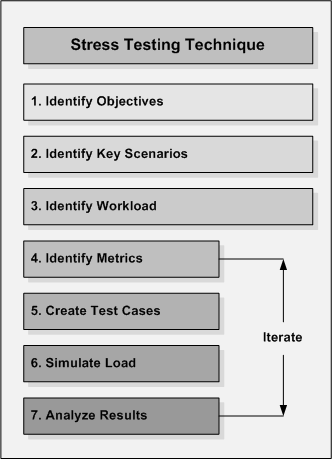
Example:

As just one example, let’s say you’re developing a new online voting platform, and you’d like for it to be able to handle potentially up to 10,000 user submissions per minute during peak load times. While developing the software, you may have performed unit tests as the code was being written, plus periodic [regression tests](http://smartbear.com/learn/automated-testing/what-is-regression-testing) to make sure you didn’t break existing functionality with each new modification as development progressed, but at what point did you [begin testing for multiple users](http://blog.smartbear.com/performance-testing/load-testing-101-essential-tips-for-developers-and-testers/) .At what point did you begin testing the program to accept hundreds or even thousands of overlapping field entries, form submissions, and other commands.

**Stress Testing:**

Stress testing is a type of *performance testing* focused on determining an application's robustness, availability, and reliability under extreme conditions. The goal of stress testing is to identify application issues that arise or become apparent only under extreme conditions.

Stress testing is a generic term used to describe the process of putting a system through exertion or stress. The Stress test can combine aspects of performance testing and security testing. An example in case is the process of stress testing firewalls which require concurrent system loads combined with actual attacks against the component. Stress testing is typically used to benchmark a systems performance to determine a systems upper performance limit and to verify how a system recovers when the workload is reduced.



**Endurance Testing:**

It is a type of non-functional testing.

It is also known as Soak testing.

Endurance testing involves testing a system with a significant load extended over a significant period of time, to discover how the system behaves under sustained use. For example, in software testing, a system may behave exactly as expected when tested for 1 hour but when the same system is tested for 3 hours, problems such as memory leaks cause the system to fail or behave randomly.

The goal is to discover how the system behaves under sustained use. That is, to ensure that the throughput and/or response times after some long period of sustained activity are as good as or better than at the beginning of the test.

It is basically used to check the memory leaks.

Example:

Below is given the real time example of Endurance Testing and hope after seen these real time example your knowledge on Endurance Testing should definitely be increased.

For example in the closing days of bank we have continues load on that days so we always test the banking application by keeping in mind the Endurance Testing.

So in Banking Application we always test how much a system/application can sustain the continuous expected load during the closing days of Bank, so it is Endurance Testing Example done on real time Banking Environment.

**Spike Testing:**

Spike testing is a type of load test.  The object of this type of performance test is to verify a system's stability during bursts of concurrent user and or system activity to varying degrees of load over varying time periods.

Example:

A fire alarm goes off in a major business center -all employees evacuate.  The fire alarm drill completes and all employees return to work and log into an IT system within a 20 minute period A new system is released into production and multiple users access the system within a very small time period. A system or service outage causes all users to lose access to a system.  After the outage has been rectified all users then log back onto the system at the same time.

Spike testing should also verify that an application recovers between periods of spike activity.

Testing Performance offers Load Testing and Performance Testing services.

**Volume Testing:**

It is a type of non-functional testing.

Volume testing refers to testing a software application or the product with a certain amount of data. E.g., if we want to volume test our application with a specific database size, we need to expand our database to that size and then test the application’s performance on it.

“Volume testing” is a term given and described in Glen ford Myers’ *The Art of* [*Software Testing*](http://istqbexamcertification.com/what-is-a-software-testing/), 1979. Here’s his definition: **“**Subjecting the program to heavy volumes of data. The purpose of volume testing is to show that the program cannot handle the volume of data specified in its objectives” – p. 113.

The purpose of volume testing is to determine system performance with increasing volumes of data in the database.

**Scalability Testing:**

It is a type of non-functional testing.

It is a type of [software testing](http://istqbexamcertification.com/what-is-a-software-testing/) that test the ability of a system, a network, or a process to continue to function well, when it is changed in size or volume in order to meet a growing need.

It is the testing of a software application for measuring its capability to scale up in terms of any of its non-functional capability like load supported, the number of transactions, the data volume etc.

**Performance testing life cycle:**

Performance engineering or SPE (Systems Performance Engineering or Software Performance Engineering) within [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering), encompasses the set of roles, skills, activities, practices, tools, and deliverables applied at every phase of the [systems development life cycle](https://en.wikipedia.org/wiki/Systems_development_life_cycle) which ensures that a solution will be designed, implemented, and operationally supported to meet the [non-functional requirements](https://en.wikipedia.org/wiki/Non-functional_requirement) for performance (such as [throughput](https://en.wikipedia.org/wiki/Throughput), [latency](https://en.wikipedia.org/wiki/Latency_(engineering)), or [memory](https://en.wikipedia.org/wiki/Computer_memory) usage).



Performance is important to the success of a project, many software products fail to respond fast enough to user requests or to handle a certain amount of parallel business transactions. This is because nowadays projects are result-oriented where the focus is on functionality to be implemented. Such projects do not pay high attention to application performance because it still does not have the particular importance that unit testing. The problem of neglecting performance is that performance issues often do not emerge until an application is put into production, where it is likely to suffer the consequences of a performance failure. To avoid such situations, performance management should be integrated into an application’s lifecycle from the beginning. Amongst others, performance testing is an important part when managing performance and therefore needs to be integrated very early in an application‘s lifecycle in order to extensively test an application for its ability of reaching performance objectives before it is deployed to a production environment and exposed to real users.

The following points have to be considered and clarified: who should be responsible for conducting performance tests? how should performance testing be carried out throughout an application‘s lifecycle. Which types of performance tests should be considered? Examples can be load tests, stress tests, or capacity tests at which stages should performance testing be conducted and in which scope? Which dependencies does performance testing have to other activities or artifacts. Expectations documentation and integration of all results into a particular configuration management system quick wittedness and high motivation creativity, flexibility and high analytical skills communicational skills and the ability to work independent.

**Performance test management tools:**

Test management tools are used to store information on how testing is to be done, plan testing activities and report the status of quality assurance activities. The tools have different approaches to testing and thus have different sets of features.