**UNIT-IV**

* **Performance Testing**

Performance testing is [testing](http://www.wikipedia.org/wiki/Software_testing) that is performed, to determine how fast some aspect of a [system](http://www.wikipedia.org/wiki/System) performs under a particular workload.

It can also serve to validate and verify other [quality](http://www.wikipedia.org/wiki/Quality) [attributes](http://www.wikipedia.org/wiki/Attribute) of the system, such as [scalability](http://www.wikipedia.org/wiki/Scalability), [reliability](http://www.wikipedia.org/wiki/Reliability) and resource usage.

Performance testing is a subset of [Performance engineering](http://www.wikipedia.org/wiki/Performance_engineering), an emerging [computer science](http://www.wikipedia.org/wiki/Computer_science) practice which strives to build performance into the design and architecture of a system, prior to the onset of actual coding effort.

* **Factors Governing Performance Testing**

There are many factors that govern performance testing. It is critical to understand the definition and purpose of these factors prior to understanding the methodology for performance testing and for analyzing the results.

The capability of the system or the product in handling multiple transactions is determined by a factor called throughput.

Throughput represents the number of requests/business transactions processed by the product in a specified time duration.

It is important to understand that the throughput (that is, the number of transactions serviced by the product per unit time) varies according to the load the product is subjected to.



Example of latencies at various levels-network and applications

## Performance Testing Methodologies

Performance testing is a subset of performance engineering; an emerging computer science practice which strives to build performance into the design and architecture of a system, often prior to the start of the actual coding effort.

Performance testing can serve different purposes. It can demonstrate that a system meets     its performance criteria, it can compare performance between two systems and it can identify which components of the system might cause bottlenecks or have an overall impact on performance.

Performance testing can also serve to validate and verify other quality attributes of the system such as scalability, reliability and resource usage; whilst helping to tune the system to best handle real production load without performance impacts.

The most important thing to note about the different types of performance testing (SVP - Stress, Volume and Performance) is that it is not trying to find functional defects. SVP needs to be able to execute end-to-end business transactions using different volumes and scenarios to determine how well some aspects of the system perform under any given load.

Requirements analysis in [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting [requirements](http://en.wikipedia.org/wiki/Requirement) of the various [stakeholders](http://en.wikipedia.org/wiki/Stakeholder_%28corporate%29), analyzing, documenting, validating and managing software or system requirements.

## Test Cases

1. List of operations or business transactions to be tested
2. Steps for executing those operations/transactions
3. List of product, os parameters that impact the performance testing and their values
4. Loading pattern
5. Resource and their configuration
6. The expected results
* **Automated Test Performance**
1. TestComplete searches for the tested object (tree view control) in the system. If the object does not exist at that time, TestComplete waits for it.
2. After TestComplete locates the object in the operating system, it determines its type and adds the appropriate methods and properties to the corresponding test object.
3. After TestComplete finds the method specified in the automated test it calls this method and proceeds with the next automated test command.

## Executing Performance Tests

There are three types of performance test cases that are typically executed: response time, stress, and reliability testing.

 Each is intended to measure different key performance indicators. Performance tests are typically managed by specialized members of the testing and administration organizations, who have ownership of the architecture and infrastructure.

|  |
| --- |
|  Diagram of the Execute Performance Tests Process |
|  Performance Testing Process |

Execute each script for a single user to validate the health of the environment. Obtain a low user-load baseline before attempting the target user load.

Use a baseline to measure scalability by comparing results between the baseline and target loads.

* **Performance tuning**

Performance tuning is the improvement of [system](http://en.wikipedia.org/wiki/System) [performance](http://en.wikipedia.org/wiki/Computer_performance). This is typically a computer application, but the same methods can be applied to economic markets, bureaucracies or other complex systems.

The motivation for such activity is called a performance problem, which can be real or anticipated. Most systems will respond to increased [load](http://en.wikipedia.org/wiki/Load_%28computing%29) with some degree of decreasing performance.

 A system's ability to accept higher load is called [scalability](http://en.wikipedia.org/wiki/Scalability), and modifying a system to handle a higher load is synonymous to performance tuning.

Systematic tuning follows these steps:

1. Assess the problem and establish numeric values that categorize acceptable behavior.
2. Measure the performance of the system before modification.
3. Identify the part of the system that is critical for improving the performance. This is called the [bottleneck](http://en.wikipedia.org/wiki/Bottleneck).
4. Modify that part of the system to remove the bottleneck.
5. Measure the performance of the system after modification.
6. If the modification makes the performance better, adopt it. If the modification makes the performance worse, put it back the way it was.

## Benchmark Testing

Benchmark testing is the process of load testing a component or an entire end to end IT system to determine the performance characteristics of the application.

The benchmark test is repeatable in that the performance measurements captured will vary only a few percent each time the test is run.

This enables single changes to be made to the application or infrastructure in an attempt to determine if there is a performance improvement or degradation.

Benchmark testing can combine aspects of security testing.- An example in case is benchmark testing  firewalls.

This requires system and or user loads combined with security violations concurrently executed against the component to determine its performance benchmark.

The goals of benchmark testing typically fall into two categories;

To test the system to measure how a change affects its performance characteristics.

To test and tune the system to reach a performance requirement or service level agreement  (SLA). In this case a series of benchmark tests are conducted in conjunction with iterative cycles of performance tuning.

## Capacity Planning

Capacity Planning is the process of determining what type of hardware and software configuration is required to meet application needs.

 Capacity planning, performance benchmarks and validation testing are essential components of successful enterprise implementations.

Capacity [planning](http://blogs.hexaware.com/quality-assurance-and-testing-services/capacity-planning-in-performance-testing-2/) is an iterative process. A good capacity management plan is based on monitoring and measuring load data over time and implementing flexible solutions to handle variances without impacting performance.

The goal of capacity planning is to identify the right amount of resources required to meet service demands now and in the future. It is a proactive discipline with far-reaching impact, supporting:

* **Performance Testing Tools**
* Apache JMeter
* NeoLoad
* LoadRunner
* LoadUI
* WebLOAD
* WAPT
* Loadster
* LoadImpact
* Rational Performance Tester
* Testing Anywhere
* OpenSTA
* QEngine (ManageEngine)
* Loadstorm
* CloudTest
* Httperf
* **Process for performance testing**
* In [software engineering](http://en.wikipedia.org/wiki/Software_engineering), performance testing is in general [testing](http://en.wikipedia.org/wiki/Software_testing) performed to determine how a [system](http://en.wikipedia.org/wiki/System) performs in terms of responsiveness and stability under a particular workload.
* Performance testing is a subset of [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering), an emerging [computer science](http://en.wikipedia.org/wiki/Computer_science) practice which strives to build performance into the implementation, design and architecture of a system.
* **Regression Testing**
* **What is regression testing**

Regression testing is done to ensure that enhancements or defect fixes made to the software works properly and does not affect the existing functionality.



* **Types of Regression Testing**

There are two types of regression testing in practice

1. Regular regression testing
2. Final regression testing

|  |
| --- |
|  |
|  |
|  |

* **When Do in Regression Testing**
* Rerunning the previously conducted tests
* Comparing current results with previously executed test results

This is a continuous process performed at various stages throughout the software testing life cycle. A best practice is to conduct regression test after the [sanity or smoke testing](http://www.softwaretestinghelp.com/smoke-testing-and-sanity-testing-difference/) and at the end of functional testing for a short release.

To conduct effective testing, regression [test plan](http://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/) should to be created. This plan should outline the regression testing strategy and exit criteria.

* **How to do regression testing**

The failure of regression testing can only be found very late in the cycle or found by customers . having a well defined methodology for regression can prevent such costly misses.

* **Sanity Testing**
Sanity testing is cursory testing, and performed whenever cursory testing 'is' sufficient to prove the application is functioning according to specifications.

Sanity testing is a subset of regression testing. It normally includes a set of core tests such as basic GUI functionality to demonstrate connectivity to the database, application servers, printers, etc.



**ONE MARK QUESTIONS**

1. The capability of the system or the product in handling multiple transactions is determined by a factor called-------------------
2. **Throughput** b. performance c. factors d. response
3. The performance testing wherein competitive products are compared is-----------------
4. **Benchmarking** b. throughput c. planning d. none
5. The exercise to find out what resources and configuration are needed is called----------------
6. **Capacity planning** b. benchmarking c. throughput d. a and b
7. The process of removing some unwanted values set is called--------------
8. **Noise removal** b. process removing c. noise process d. none
9. ---------------- is done to ensure that enhancements or defect fixes made to the software works properly and does not affect the existing functionality
10. **Regression testing** b. system testing c. final regression d. none
11. Regression testing follows-------------- technique
12. **Selective re-testing** b. regression c. regular regression d. none
13. A ---------------- is done to validate the final build before release
14. **Final regression testing** b. regular regression c. selective d. regression
15. ------------------testing should focus more on the impact of defect fixes than on the critically of the defect itself
16. **Regression** b. system c. load d. stress
17. Resetting a test case is nothing but setting a flag called------------------
18. Not run b. execute again c. **a and b** d. none
19. For regression testing all priority test cases are return is----------------
20. **Regress all** b. regress changes c. random regression d. none
21. Random test cases are selected and executed for this regression methodology is-------------
22. **Random regression** b. regression changes c. regress all d. none
23. For regression testing based on this priority test cases are run based on the availability of time is----------
24. **Priority based regression** b. regress all c. regress changes d. none
25. SDLC stands for--------------
26. **Software Development Life Cycle** b. system development life cycle c. software design life cycle d. none
27. -----------------can be performed irrespective of which test phases the product is in
28. **Regression testing** b. integration testing c. stress testing d. white box testing
29. A --------------------- configuration denotes that with anything less than this configuration , the product may not even work
30. **Minimum required** b. maximum required c. expected required d. none
31. The results of performance tuning are normally published in the form of a guide called-------------a. **performance tuning guide** b. performance benchmarking c. performance tuning d. none
32. There are two types of requirements performance testing focuses on-----------------
33. Generic requirements b. specific requirements c. system requirements

d. **a and b**

1. ------------- can be defined as the delay between the point of request and the first response from the product
2. **Response time** b. request time c. system time d. a and b
3. ---------------- is delay caused by the application , operating system and by the environment that are calculated separately
4. **Latency** b. request c. response d. wait
5. Which is the load testing tool-------------
6. **Silk performer** b. winrunner c. silktest d. QA partner

**ANSWER THE FOLLOWING QUESTIONS**

5 MARKS

1. What is performance testing?
2. Write about methodology for performance testing?
3. Write the process for performance testing?
4. What is regression testing?
5. When to do regression testing?

8 MARKS

1. Describe factors governing in performance testing?
2. Explain methodology for performance testing?
3. Explain how to do regression testing?
4. Explain challenges in performance testing?
5. Write best practices in regression testing?