**DAY -1**

**Phases of performance testing**

1. Identify Test Environment
2. Identify Performance Metrics
3. Plan and Design Performance Tests
4. Configuring test ENV
5. Implementing test Design
6. Executing Tests
7. Analyse report and Retest.
8. **Identify the Testing Environment.**

Identifying the hardware, software, network configurations and tools available allows the testing team to design the test and identify performance testing challenges early on. [Performance testing environment options](http://www.kualitatem.com/blog/performance-test-environment-setup) include:

* Subset of production system with fewer servers of lower specification
* Subset of production system with fewer servers of the same specification
* Replica of productions system
* Actual production system

1. **Identify Performance Metrics.**

In addition to identifying metrics such as response time, throughput and constraints, identify what are the success criteria for performance testing.

1. **Plan and Design Performance Tests.**

Identify performance test scenarios that take into account user variability, test data, and target metrics. This will create one or two models.

1. **Configure the Test Environment.**

Prepare the elements of the test environment and instruments needed to monitor resources.

1. **Implement Your Test Design.**

Develop the tests.

1. **Execute Tests.**

In addition to running the performance tests, monitor and capture the data generated.

1. **Analyse Report and Retest.**

Analyse the data and share the findings. Run the performance tests again using the same parameters and different parameters.

**DAY -2**

**Phase – 3: Plan and Design Performance Tests**

**Test Plan Document**

1. **Objectives:** It varies depending on the system under test.
   1. Verify application can handle user load.
   2. Response times within limits.
   3. Identify performance issues.
2. **Scope:** It should be clearly defined in the P.T plan
   1. Components of system.
   2. Types of performance test.
   3. Metrics to measure.
3. **Methodology:** It depends on objective of testing and the system under test.
   1. Load test.
   2. Stress Test.
   3. Endurance Test.
4. **Test Scenario:** To simulate realistic user behaviour during testing.
   1. In e-commerce website the scenarios include.
      * + - Browser products.
          - Adding items to cart.
          - Checking out.
          - Expected load, duration and other parameters.
5. **Test Environment:** The test env should be setup to simulate the production env as closely as possible.
   1. Hardware.
   2. Software.
   3. Network infrastructure.
6. **Tools:** Tools and technologies should be identified and includes in performance test plan.
   1. Load testing tools.
   2. Monitoring tools.
   3. Profiling tools.
7. **Timeline:** For P.T project should be clearly defined.
   1. Start date.
   2. End date.
   3. Production date.

**Test Strategy Document**

1. **Test Types:** Types of P.T tests to be conducted should be defined.
   1. Load test- To test the system’s ability user load.
   2. Stress test- Handle peak load.
   3. Volume test- Test the system with large data.
2. **Test Approach:** For each type test should be defined in test strategy doc.
   1. Load test- Ramp up, steady state, ramp down.
   2. Stress test- Peak load level/duration of test.
3. **Performance Metrics:** Should be defined in the test strategy doc.
   1. Response time.
   2. Throughput.
   3. Error rate.
   4. Resource utilization.
4. **Test Data:** To be used during the testing should be defined.
   1. User data.
   2. Input data.
   3. Test config.
5. **Environment and Infrastructure:** The env and infrastructure should be defined in the test strategy doc.
   1. Hardware.
   2. Software.
   3. Network setup.

**Questions on Test Planning**

1. **How do we determine the objectives of performance testing?**
   * + - Goals and needs of the system.
       - Identify key objectives by working with stakeholders.
2. **What type of performance tests to be include?**

* Load Test – To test the system’s ability to handle certain number of concurrent users.
* Stress Test – To test the system’s ability to handle the peak loads.
* Endurance test – To test the system’s performance over extended period of time.
* Volume testing – To test the system’s performance with large amount of data.

1. **What should be included in the test scenarios for P.T?**
   * Test scenarios for P.T should be simulating realistic user behaviour on the system under the test.
2. **How should the P.T environment be set up?**
   * Simulates the production env as closely as possible.
     1. Hardware.
     2. Software.
     3. Network infrastructure.
3. **What tools** **and technologies should be used?**
   * Depends on the system under test and objectives of the testing.
   * For performance JMeter, LoadRunner, etc.
   * For monitoring nagios and
   * For profiling VMstat, yourkit.

**DAY – 4**

**Phase – 4: Configure the Test Environment**

Types of performance testing

1. Load Testing.
2. Stress Testing.
3. Spike Testing.
4. Endurance Testing.
5. Scalability Testing.
6. Volume Testing.
7. **Load Testing**

[Load testing](http://searchsoftwarequality.techtarget.com/definition/load-testing) measures system performance as the workload increases. That workload could mean concurrent users or transactions. The system is monitored to measure response time and system staying power as workload increases. That workload falls within the parameters of normal working conditions.

1. **Stress Testing**

Unlike load testing, [stress testing](http://searchsoftwarequality.techtarget.com/definition/stress-testing) — also known as fatigue testing — is meant to measure system performance outside of the parameters of normal working conditions. The software is given more users or transactions that can be handled. The goal of stress testing is to measure the software stability. At what point does software fail, and how does the software recover from failure?

1. **Spike Testing**

Spike testing is a type of stress testing that evaluates software performance when workloads are substantially increased quickly and repeatedly. The workload is beyond normal expectations for short amounts of time.

1. **Endurance Testing**

Endurance testing also known as soak testing is an evaluation of how software performs with a normal workload over an extended amount of time. The goal of endurance testing is to check for [system problems such as memory leaks](https://msdn.microsoft.com/en-us/library/ms859408.aspx). (A [memory leak](https://stackify.com/java-memory-leaks-solutions/) occurs when a system fails to release discarded memory. The memory leak can impair system performance or cause it to fail.)

1. **Scalability Testing**

Scalability testing is used to determine if software is effectively handling increasing workloads. This can be determined by gradually adding to the user load or data volume while monitoring system performance. Also, the workload may stay at the same level while resources such as CPUs and memory are changed.

1. **Volume Testing**

Volume testing determines how efficiently software performs with large projected amounts of data. It is also known as flood testing because the test floods the system with data.

**Note: All the tests should be documented.**

**DAY - 4**

If client gave particular hits, we have to decide how much users we need to meet the requirement.

Work load modelling.

Soak testing should conduct only on non-business hours.

We can conduct Load and Soak on the same day.

We can create reports on same day.

We need to split the user load, for this we need business analyst to tell how much user load need to be split B/W each user or each script.

We can also collect user load split from the Splunk, google analytics, or any analytic tools.

Plan & Design performance Tests this page takes 60% of process.

**DAY – 5**

**Creating the web application scripts and API scripts.**

Test plan > Thread group>