**Intro. Of Performance**

Performance testing should start early and continue through the whole development process , and not leaving it to the end of process

Performance testing to know how fast an app executes an operation and to know how something like database perform under a given load to see the impact to locate bottlenecks or performance issues

Performance testing is done after all functional testing is passed

Performance testing is not only about measuring speed , Performance is measured in terms of response time , throughput, reliability, scalability

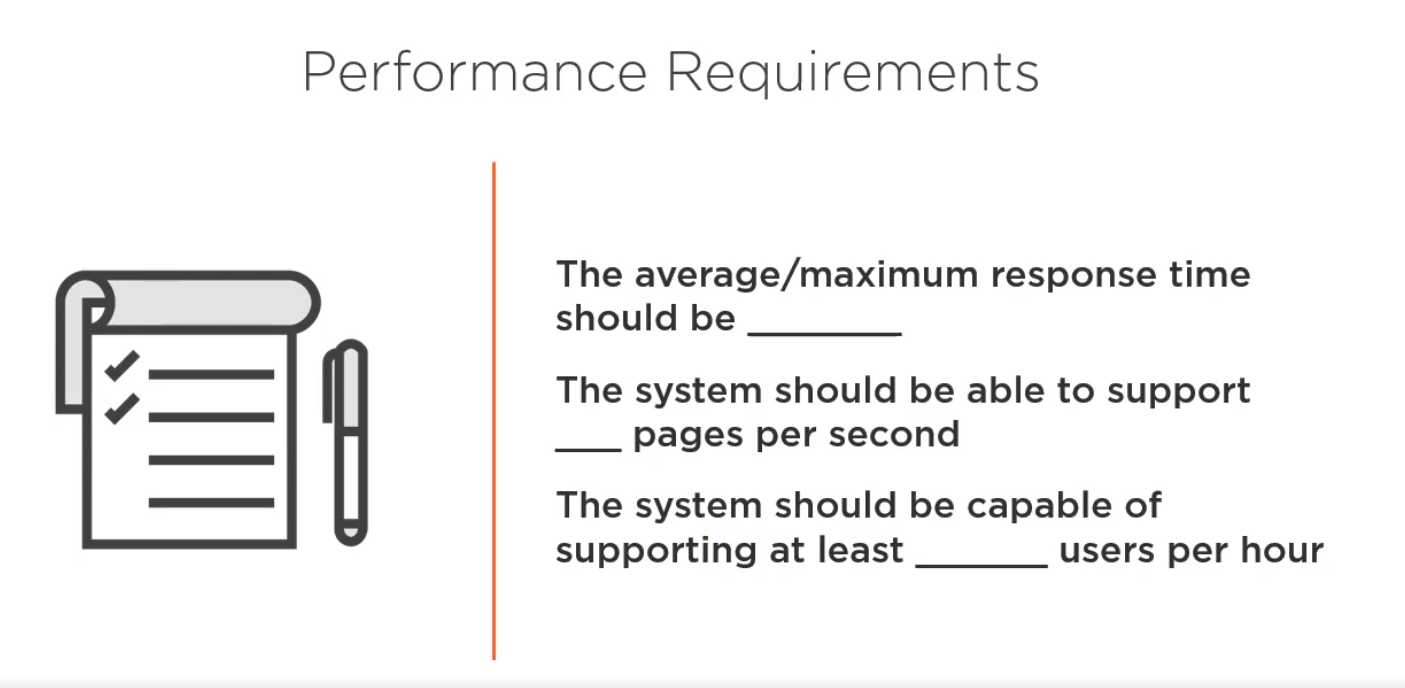
* Response time -> it is the time the app takes to respond to our request  
  ( Request time + Processing Time + Response Time) = (the time the request takes to reach the server + the time the server takes to process the request and generate a response + the time it takes the response to reach the browser)  
  Processing Time depend on complexity of Request time or server hardware   
  Response Time depend on network latency   
  **The less the better**
* Throughput -> Num of transactions (req./res.) / unit of time   
  is the number of units of work that can be handled per unit of time; for instance, requests per second, calls per day, hits per second, reports per year, etc.  
  **The more the better**
* Reliability -> how well the app detect and handle errors in terms (number of errors / number of requests)  
  **The less the better**
* Scalability -> how well the sys expands its capacity in terms of Response time, Throughput, Reliability  
  sys hy3ml eh lw ana zodt resources feh
* Think time -> is the time where user waits between each action.
* Bandwidth -> data rate supported by a network connection or interface
* Hit per second -> is the number of calls to the Webserver per second, "one Request may have many calls(Hits)"

**Type of Performance testing**

1. Stress -> test app with loads beyond normal usage to see up to which point it stays stable and responsive and see at what point the app stops working
2. Endurance -> subjected to load within its limits, but for a long duration, hours or in some cases days to see if the app has memory leaks or doesn’t properly close database or network connections for
3. Load -> behavior of app at many load level .. 2 ways increase (Num. of user or Num. of requests)
4. Spike -> subjected to sudden increment in the load beyond its maximum capacity to see if the app is robust enough to work correctly during and after the spike
5. Peak
6. Performance
7. Recovery
8. Capacity

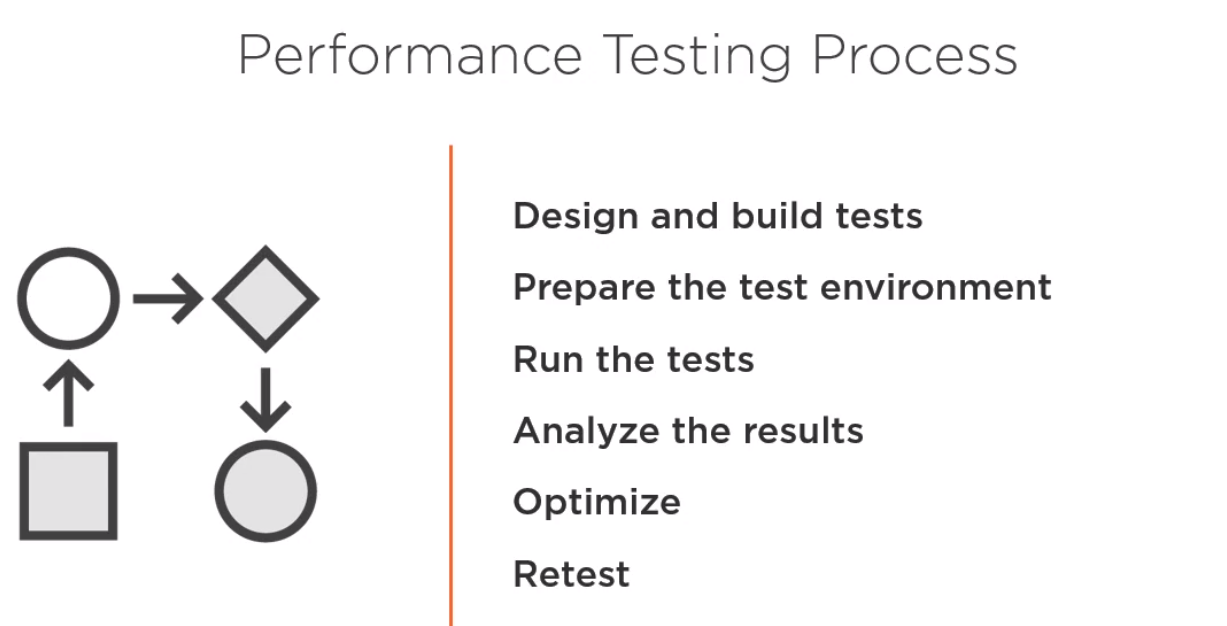
**Performance requirement**

Need to verify this with customer

****

**Performance requirement**

Should be repeated until the results is met with requirment

****

Commercial tool : LoadRunner

Open Source tool : JMeter (load testing tools)

**JMETER**