**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

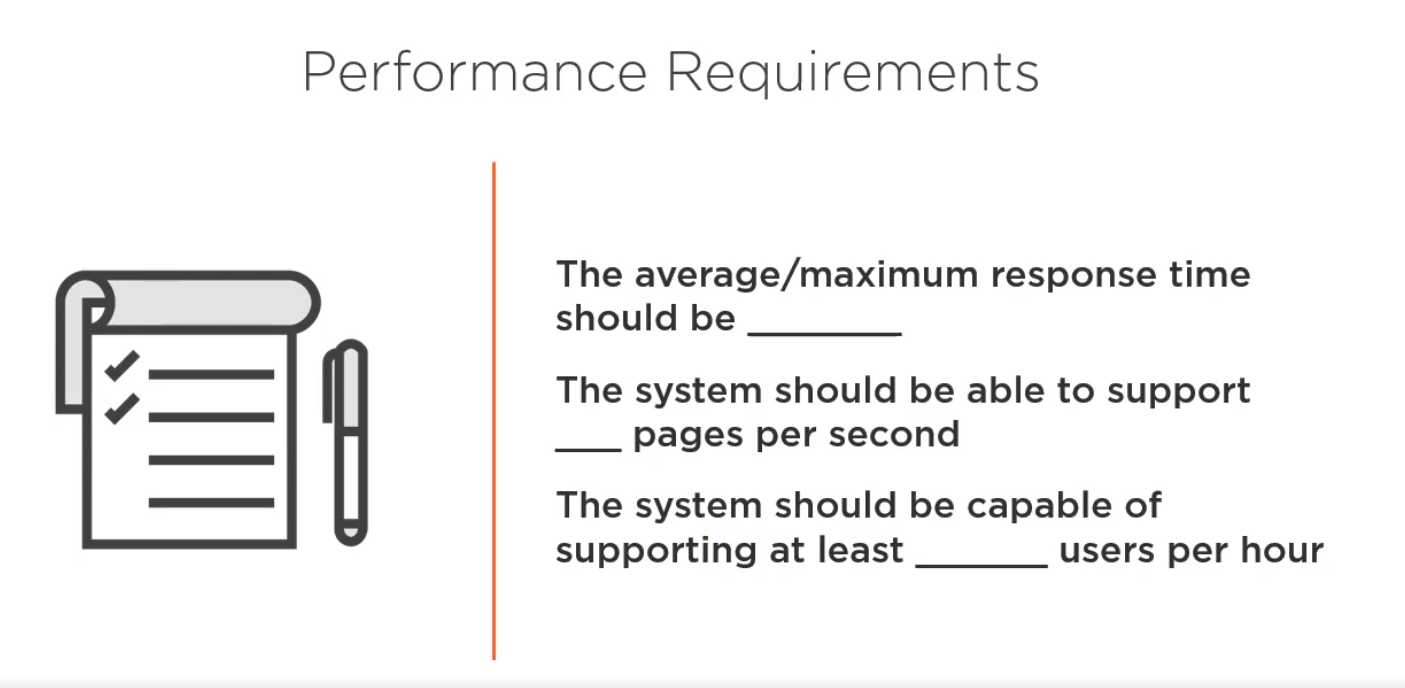
* Response time -> it is the time the app takes to respond to our request  
  ( Request time + Processing Time + Response Time) = (the time the request takes to reach the server + the time the server takes to process the request and generate a response + the time it takes the response to reach the browser)  
  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

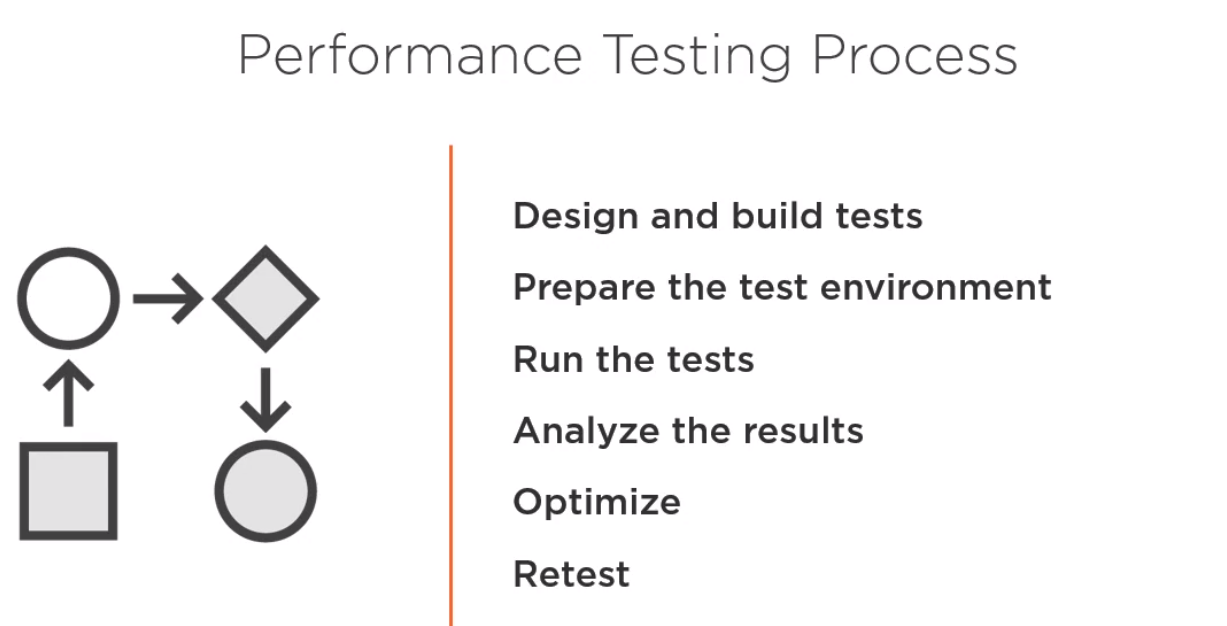
**Performance requirement**

Need to verify this with customer

****

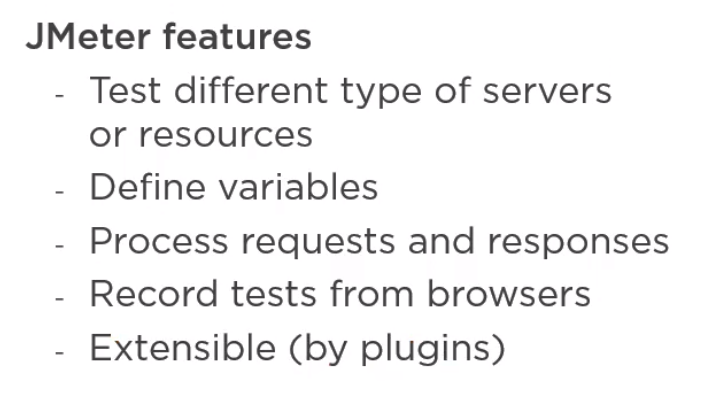
**Performance requirement**

Should be repeated until the results is met with requirment

****

Commercial tool : LoadRunner

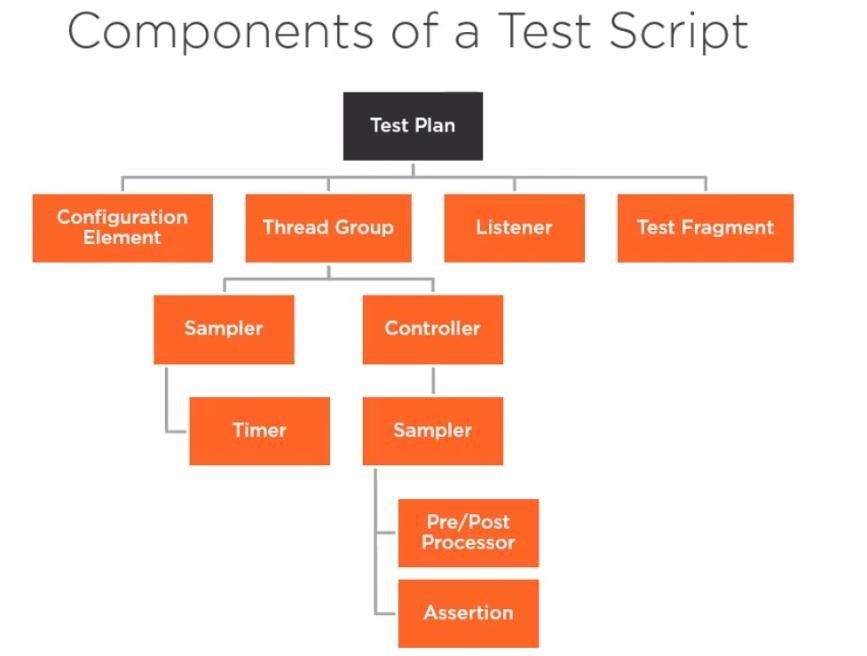
Open Source tool : JMeter (load testing tools)

**JMETER**

**Installation**

1. JDK and set Java-home environment variable
2. Download Apache Jmeter(Binaries) and extract it
3. Open bin folder then jmeter .bat
4. Plugins exist in lib/ext .. if u want to add new plugin .. download it from Jmeter plugin manager then move jar file to ext and restart Jmeter

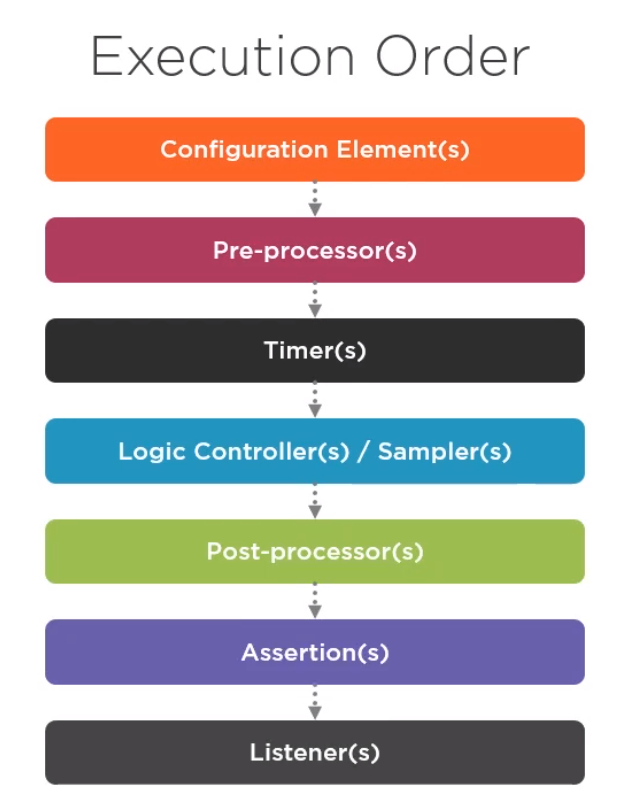
**Component of test script**



* Test Plan

1. Configuration element: set up default cof. And variables  
   can be added under any component   
   types of element:  
   1. Element allow us to define variables like : CSV data , counter, random variable   
   2. Element allow us to define conf. like : JDBC , Keystore,login  
   3. Managers allow us to conf. parameters like : http (header , cookie , cache)  
   4. Default conf. element allow us to define full conf. for request type like : http,ftp,java request
2. Thread Group: control num of users(threads) will be used to execute test  
   each thread group represent a test case   
   you can added from test plan
   1. Sampler
      1. Timer : dh hy3ml delay between requests(samplers), time user take to perform actions between pages
   2. Controller (logic controller
   3. Sampler : add http request for each page
      1. Pre/Post Processor
      2. Assertion : to validate a response is as expected  
         can be added on each sampler or each thread group
3. Listener: reports, collect info about request or reponse like : view result tree  
   can be added at any level
4. Test fragment

**Execution Order**



Last 3 will not be executed if there is no server response

**Types of Parameter:**

1. User Defined variables ${NameOfVariable}
2. CSV Data set config
3. Random variables

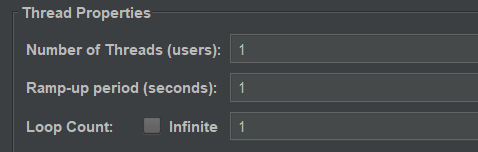
**Pre Processor & Post Processor:**

**Functions:**

There are many functions that user can use them .. can find them from Tools

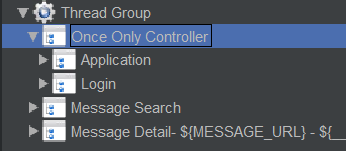
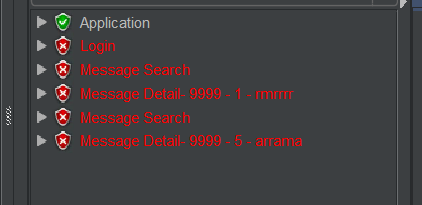
${\_\_NameOfFunction} or ${\_\_NameOfFunction(para 1, para 2,…..)}

**Number of Users:**

****

From thread group can enter number of user (all samplers will run in the same time ex 3 times Ex: login login , home home , and soon)

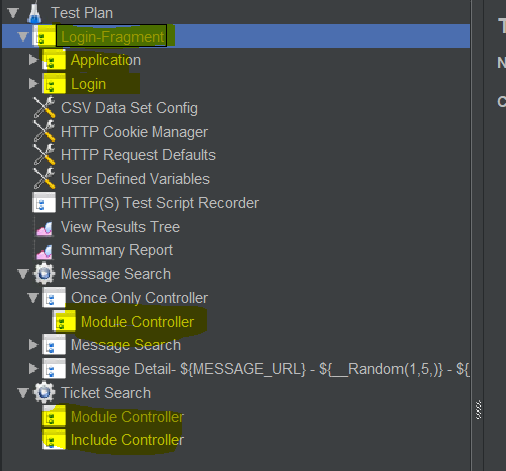
Number of loop all samplers will run how times per user (ex: 2 loop ( login , home , login, home ) per user) if u want for example to run login one time , u can put ur samplers inside ‘Only Once Controller’

**Test Fragment:**

If user want to create more than one thread group and there are some shared samplers .. so user should use test fragment and there are 2 types

1. Module controller (in the same test plan)
2. Include controller (external JMeter file)



To monitor requests u can use listeners but for server u can use PerfMon plugin allow to monitor resources on server like CPU, Memory, disk

And there is HTML report

