Performance Testing Course : Introduction to performance testing methodology

Author : Hien HOANG

Agenda:

1. Why performance testing life cycle(PTLC) is important?
2. Phase of PTLC and its details.

Why we need a PTLC?

Problem:

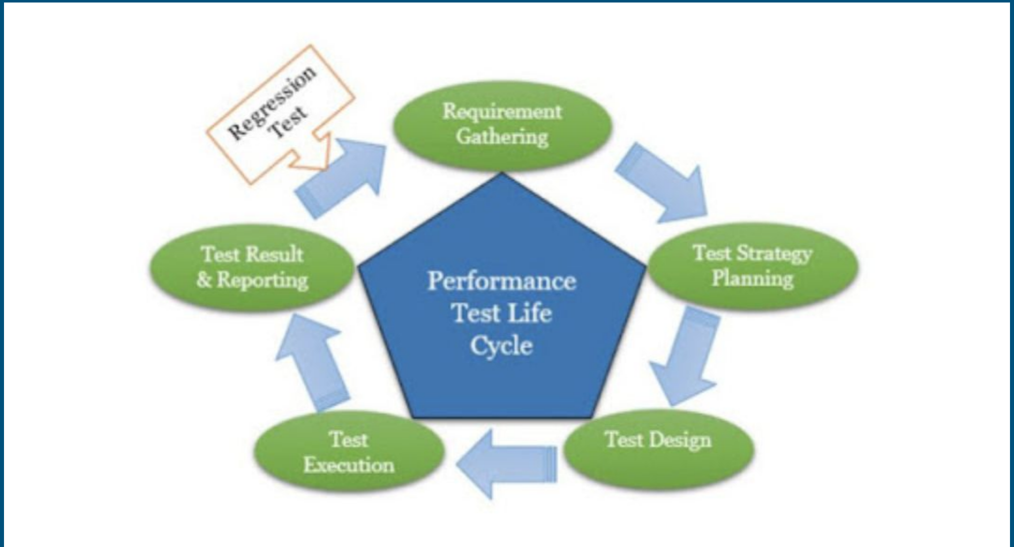
- How to measure performance of

application?

- What would be performance metrics?
- What system to be involved? On what

Basics?

=> Process.



Phase of PTLC?

1. Performance gathering
   - Main goal : Understand client's expectation and conclude requirements?
   - Accountability: Performance Test Lead/Manager
   - Deliverable : Signed-off Non-functional Requirement Document.

Example of NFRD:

https://drive.google.com/open?id=13i8jjpJiwbvm82P1cVZOB3fnDFUcDwKq

Phase of PTLC?(continued)

2. Test strategy:

- Main goal : Prepare test strategy as per non-functional requirement.
- Accountability: Performance Test Lead/Manager.
- Deliverable: Signed-off Non-functional Test Plan Document.

Example:

https://drive.google.com/open?id=159NXKUT4O9xfiD1j6iUuZhNG0Eq97AHP

Phase of PTLC?(continued)

3. Test design:

- Main goal : Create test script as per non-functional test plan.
- Accountability: Performance Test Team Member/Analyst/Lead/Manager.
- Deliverable: Performance Test Scripts.

Phase of PTLC?(continued)

4. Test execution:

- Main goal : Run the tests mentioned in Non-Function Test Plan using Test Scripts.
- Accountability: Performance Test Team Member/Analyst.
- Deliverable: Test Results.

Phase of PTLC?(continued)

5. Reporting and Recommendation:

- Main goal : Analyse tests , publish Test Report and provide recommendations.
- Accountability: Performance Test Lead/Manager.
- Deliverable: Signed-off Non-Functional Test Report.

Example:

https://drive.google.com/open?id=159NXKUT4O9xfiD1j6iUuZhNG0Eq97AHP

Requirement Gathering in-depth

TOUGHEST PHASE OF PERFORMANCE TESTING.

- Collect correct performance testing requirement.
- Conclude proper client's expectation.
- Sometimes, client did not provide suitable information => Performance test becomes showcase.

Requirement Gathering in-depth(continued)

UNDERSTAND APPLICATION ARCHITECTURE:

- Setup multiple calls and meetings with stakeholders.
- Collect as much as possible about application and its architecture

Example:

https://drive.google.com/open?id=1af97K5g_2cWSoedmvXBZnzYHFs2U04IZ

https://drive.google.com/open?id=18h3ssUodT8lEk4y3gX5qP5MXpOt3sRpa

https://drive.google.com/open?id=1Md1lRZhaA7KSb3OM2mgCNiCceyiozMjL

Requirement Gathering in-depth(continued)

Showcase:

An e-commerce website with 3-tier architecture needs to be tested. A performance tester (name Thanh) set-up some initial meetings with project stakeholders to understand the application architecture and gather the non-functional requirement. After understanding the application architecture, Thanh got the following requirements:

1. The application should be very fast.
2. The response time of the application should be quick.
3. The web server performance should be as high as possible.
4. The application should support many users.
5. The application should not fail when a sudden load comes during sale and offer periods.
6. The application should run without any failures for a long duration.

Test strategy in depth

- Map non-functional requirement with non-functional test.
- In the test plan, specify testing tool to be used, types of tests to be conducted, business scenarios to be included, effort required for performance test, entry and exit criteria for conducting the test,test data and test environment,etc…
- Note : RAID(Risk, Assumption, Issue and Dependency) are VERY important section in the test plan that save tester in critical time when project start bombarding and blaming tester in case of delay.

Test design

- Scripts should be prepared with sufficient test data in order to simulate multiple user loads on the system.
- Note: Many companies provide dev environment to create the script.

  => You may face some issues like failure of the scripts due to changes in the code or configuration which you can rectify during the script validation phase.

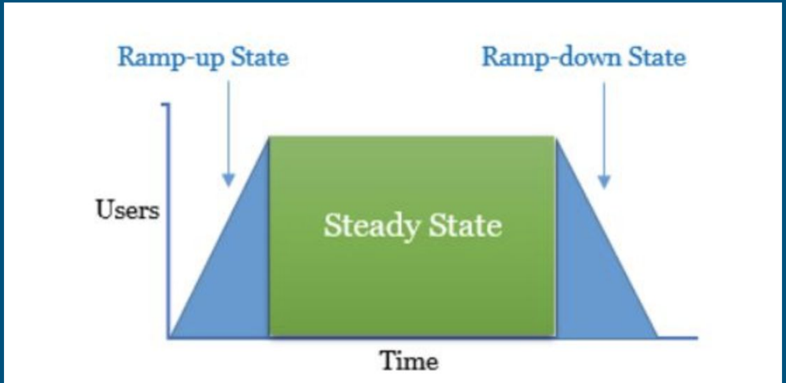
  BUT tester could work parallel with functional team, initial functional bug identification, lower down end-time pressure.

Test execution

WORKLOAD MODELING:

- When scripts are created and validated, the planned tests are executed and the SUT is certified during this phase BUT before stating the actual performance test, first thing to be considered is “Workload modelling or scenario creation”

=> WORKLOAD? : Distribution of load across the identified scenario at a given time.



Test execution(continued)

Why WORKLOAD is needed?

- It helps to study behavior of system under various identified workload model.

How to design WORKLOAD?

- By predictability(stress test), repeatability(regression load test) and scalability(stress/breakpoint test)
- It is important to collect relevant data to create effective workload model.
  - Number of concurrent users
  - Total transactions to be achieved.

Test execution(continued)

Overall Response Time (time to complete one iteration) is unknown, so either tester assume the overall response time or needs to execute each script with one user and get actual response time.

Thinktime is the time between 2 pages which shows that user halt on the previous page to read the page content or fill the form or wait for the whole page loading etc…

Pacing is the waiting time between each iteration. It could be calculated by:

Pacing = (No. of Users * No. of Transaction/TPS) - (Overall Response Time+Total ThinkTime)

where Total ThinkTime = Individual ThinkTime * (No. of Transactions - 1)

Test execution(continued)

Important notes:

- Check whether Load Generator/Injector and Controller have sufficient memory/disk space/etc…
- Reboot LG and controller if feasible.
- Get confirmation from dev whether latest code is deployed in performance test environment.
- Verify the script by running a smoke test before starting the actual load.(unit test)
- Verify all the required files like test data, etc.. are properly placed and configured(calibrage test)
- Restart web/application/database server before the test.
- Clear server log before the test.
- Conduct a quick health check to see if environment is stable.
- Verify whether required monitors are up and running.
- Validate run-time settings.

Reporting and recommendations.

- Test analysis and recommendations are documented in understandable format and presented to project stakeholders.
- It is better if result after each test is published so stakeholders could be aware of all situation.
- It is VERY important that the test report should be presented in a detailed manner with explanations for the bottlenecks identified during PTLC.

IMPORTANT NOTES:

- Verify whether all related NFRs meet or not and mark it with result PASSED,PARTIAL PASSED or FAILED.
- Test summary should be in simple language with all the key observations listed in points.
- Test report must be reviewed by performance test lead/manager.
- Recommendations should be related to the application performance improvement.