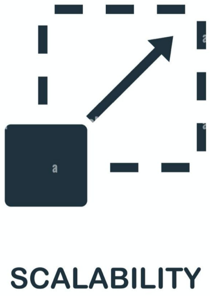
Performance Testing

/thoughtworks

Table of contents

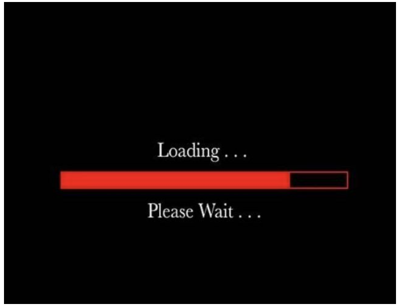
Introduction to Performance Testing

- **Performance testing** is a non-functional software testing technique that determines how the *speed* ,*stability* and*scalability* of an application holds up under a given *load*.



Performance Problems

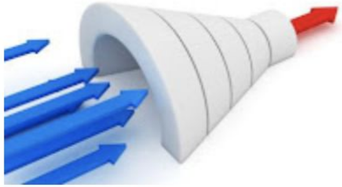
Long load Time



Poor Response Time

Poor Scalability

Bottlenecking



Why do Performance Testing?

- Performance Testing uncovers what needs to be improved before the product goes to market.
- Provides better quality, more scalable code.
- To ensure that system meets performance expectations such as response time throughput etc under given levels of load.
- Identifies problems early on before they become costly to resolve.
- Prevents revenue and credibility loss due to poor application performance.
- To find out performance bottleneck.
- To check the behavior of the application at average and peak user load.
- To know maximum load handling capacity of the system.(System Breakpoint)

POLLS

When is it required?

- **Design Phase**:

Pages containing lot of images and multimedia for reasonable wait times. Identify the types of content causing slowdown.

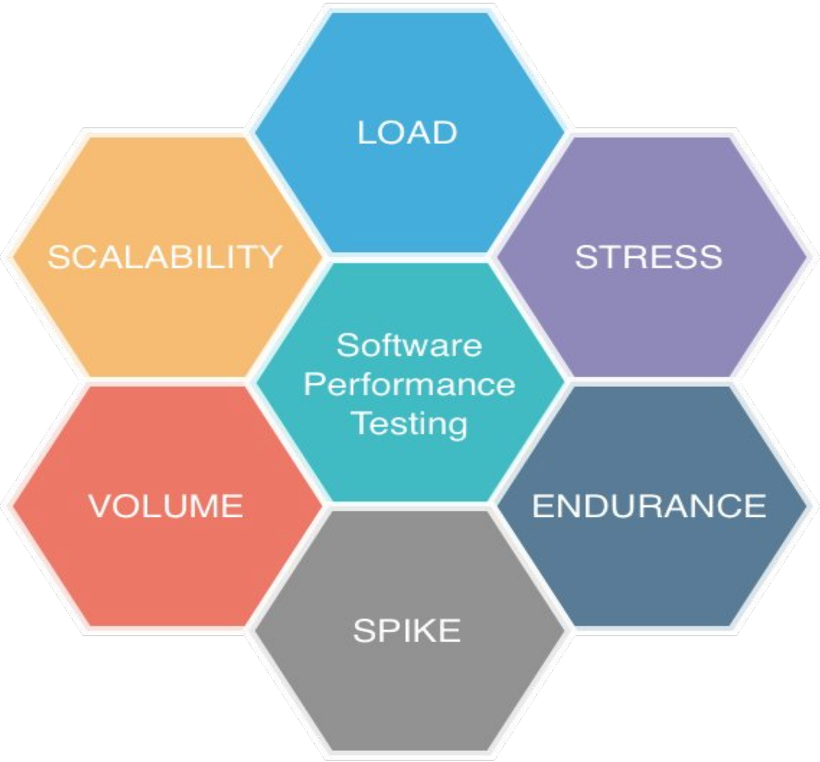
- **Development Phase:**

Checking result of individual pages and processes, looking for breaking points and bottlenecks.

- **Deployment Phase:**

Identify the minimum software and hardware requirements for the application.

Types of Performance Testing



Types of Performance Testing

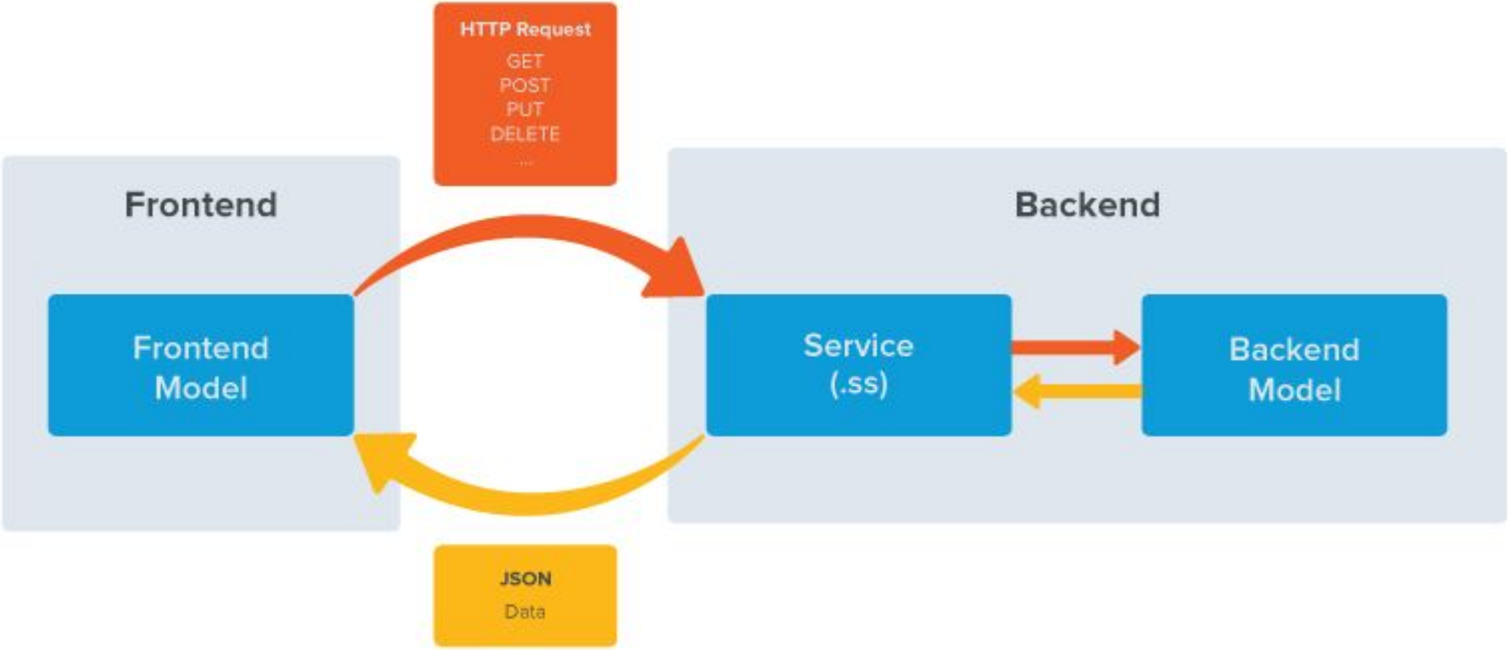
- **Load testing –** checks the application’s ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.
- **Stress testing –** involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.
- **Endurance testing –** is done to make sure the software can handle the expected load over a long period of time.
- **Spike testing –** tests the software’s reaction to sudden large spikes in the load generated by users.
- **Volume testing** – Under Volume Testing large no. of. Data is populated in a database and the overall software system’s behavior is monitored. The objective is to check software application’s performance under varying database volumes.
- **Scalability testing** – The objective of scalability testing is to determine the software application’s effectiveness in “scaling up” to support an increase in user load. It helps plan capacity addition to your software system.

POLLS

How to choose the right performance testing tool

- Budget & Licensing costs
- Protocols
- No. of Concurrent Users
- Result generation option
- Tool support
- Recording option
- Integration with other tools
- Git Friendly

Frontend Testing Vs Backend Testing



Performance Testing Tools

- JMeter
- LoadRunner
- Locust
- Gatling

Front-End Performance Testing Tools

- Lighthouse
- WebPageTest