

# Performance Testing

## Introduction

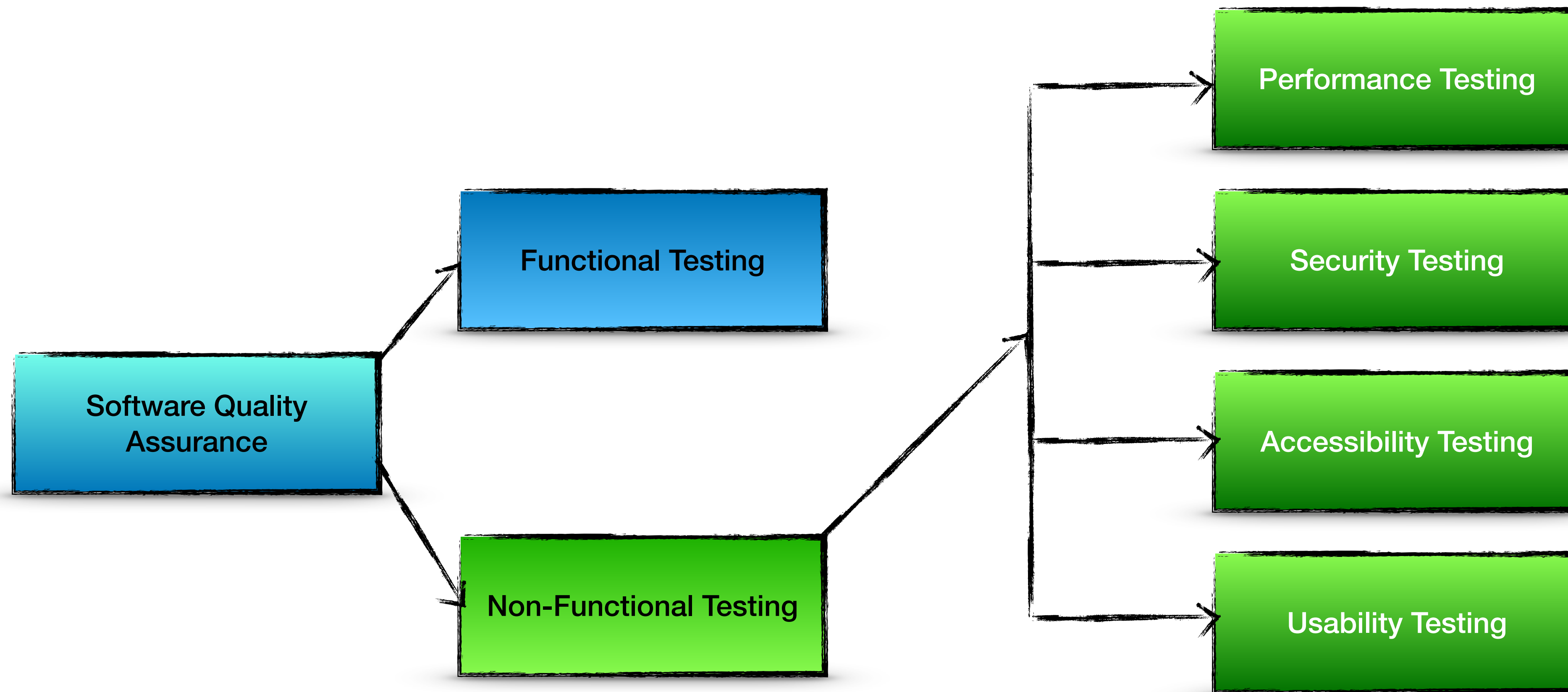
Alex Volkov 05/15/2022

# Performance testing

- What is Software Testing?
- What is Performance Testing?
- Why Performance Testing is required?
- Type of Performance Testing
- Performance Testing Tools

# What is Software Testing

- Software Testing - is a Software Quality Check Procedure that assures that the developing software corresponds to the planned business requirement.

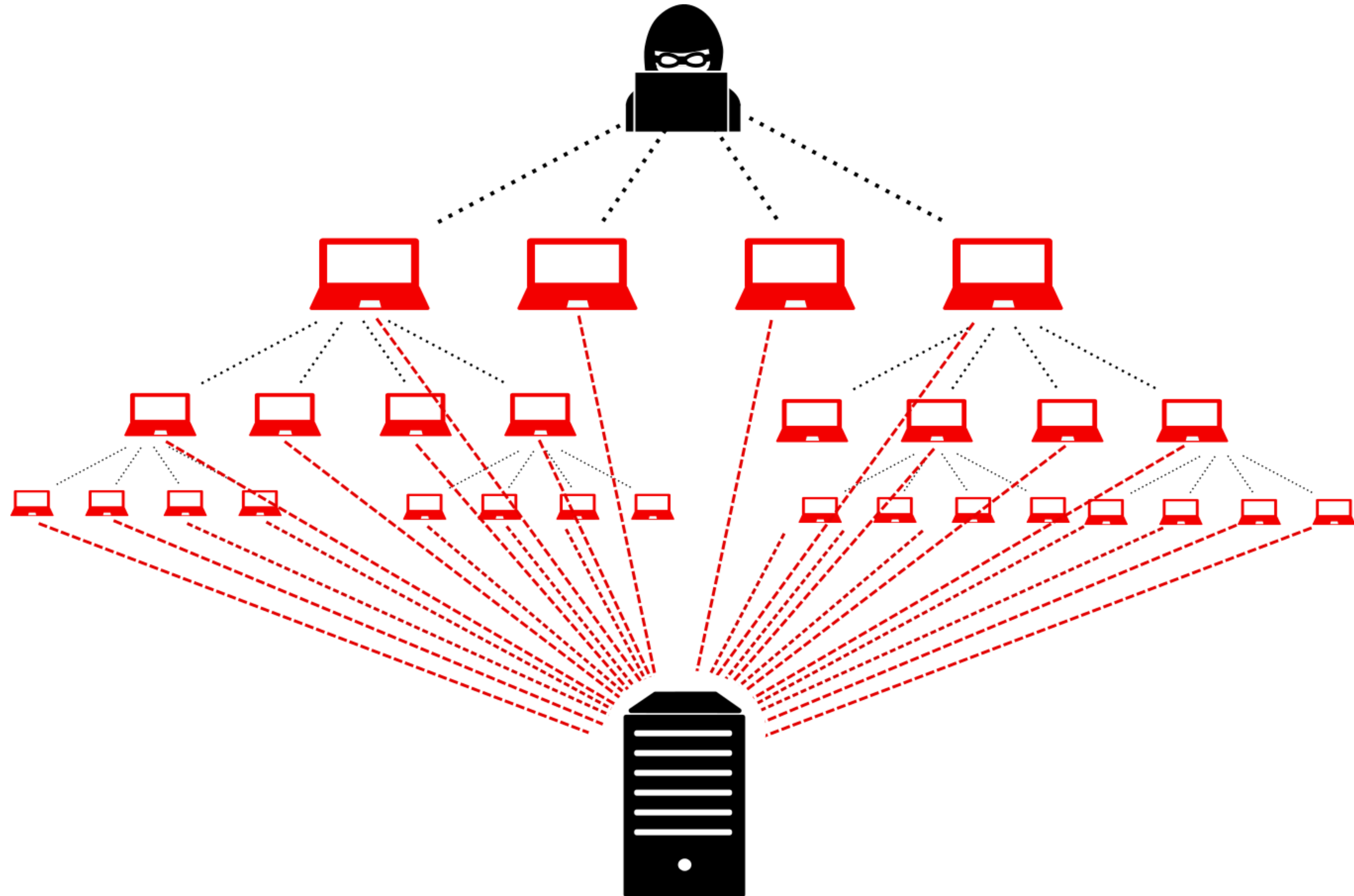


# Performance Testing

- Performance testing is a systematic testing approach to validate the performance of an application under a certain load
- Performance Testing is a Non-Functional Testing in which the preference is given to the application/system load, response timing and systems resource utilization
- The example of Functional vs. Performance testing:
- When a customer places an online order:
  - the correct order number and confirmation of the transaction is a functional testing
  - the application responsiveness (per time unit) and ability to perform multiple correct transactions at the same time is a performance testing



# Performance testing

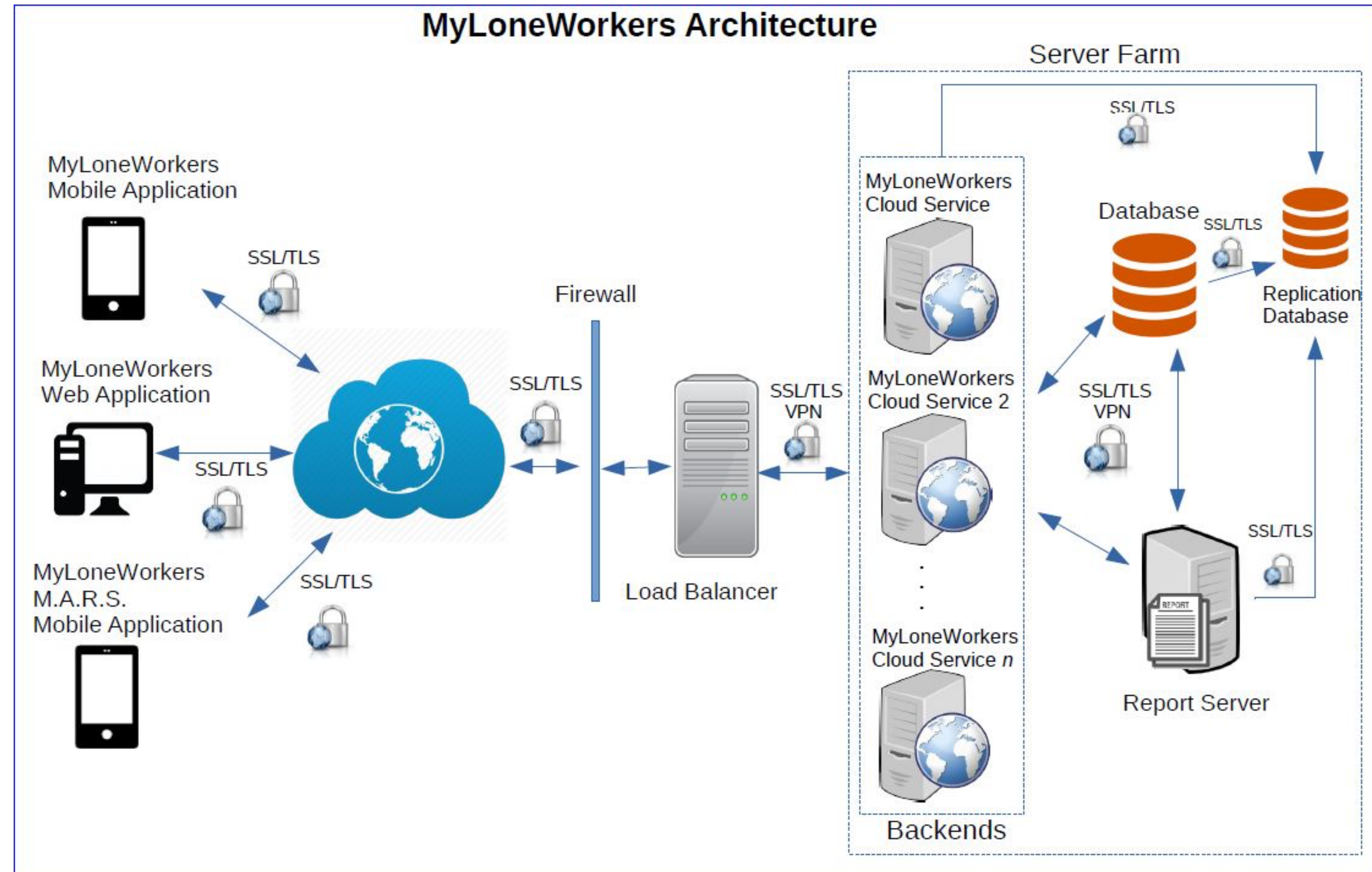


# Need of Performance Testing

- To measure the responsiveness of the website/system
- To know the maximum load handling capacity of a website/system
  - System Break-point
- To check the behavior of the website/system at average and peak user load
- To understand the system resource utilization
- To find out the performance bottleneck

# Categories of Performance Tests

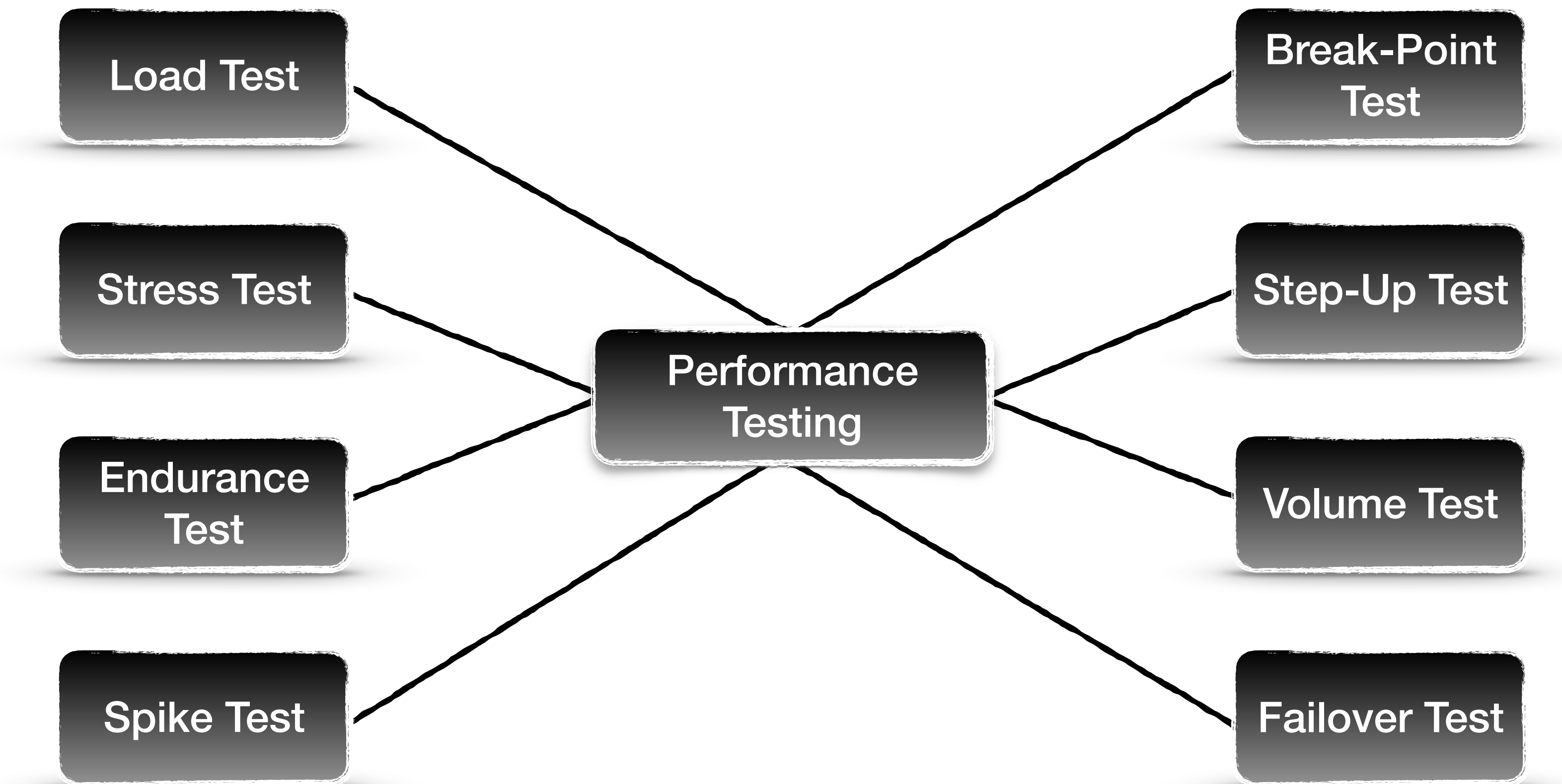
- **Back-end**
  - Unit test
  - Integrated Test
- **Front-end**
  - User Performance Experience
  - Manual
  - Automated





# Types of Performance Testing

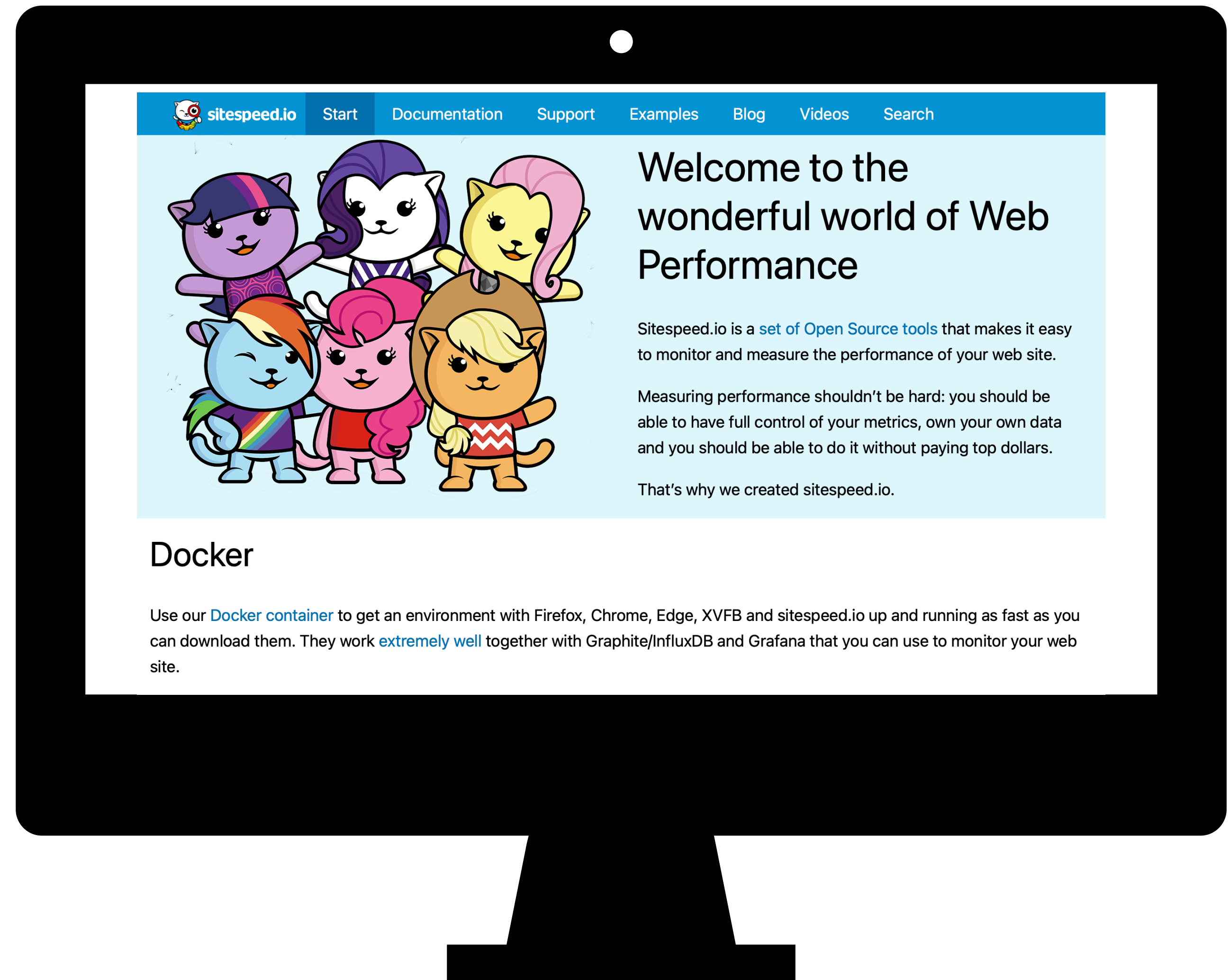
## Back-End



# Types of Performance Testing

## Front-End

- **Manual tests**
  - Extensions
  - WebTools
  - Standalone tools
- **Automation test**
  - Standalone tools



# Performance Testing Tools

## Some Examples

- Apache JMeter
- LoadRunner
- K6
- Gatling
- LoadNinja
- WebLoad
- Sitespeed.io
- BlazeMeter
- NeoLoad
- LoadView
- StomForge
- Rational Performance Tester
- Lighthouse
- ReadyAPI

# Performance Testing

## Engineering vs. Testing

Alex Volkov 05/15/2022

# Performance Engineering

## Understanding

- A systematic approach of constructing the software that meets the performance objectives
- Driven by business requirements
- To design a tuned software application at the early stage of the software development life cycle
- Reduces the last-minute application tuning activities





# Performance Engineering

## Needs

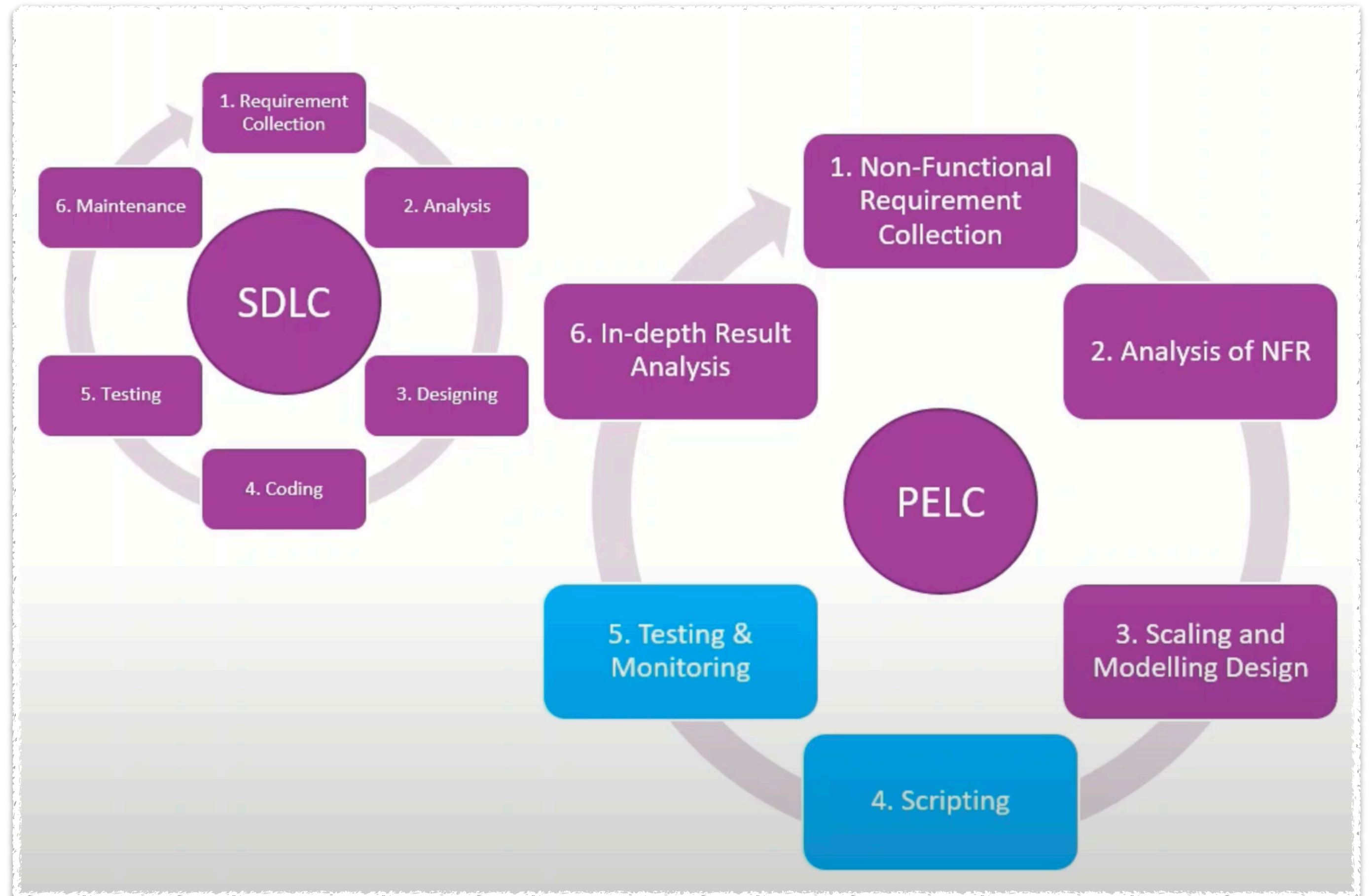
- Reduces the need to rework and refactor the application
- Cost saving
- Improve the performance of legacy system
- Implementation of DevOps
- Enhance End-to-End application's performance optimization



# Performance Engineering

## Performance Engineering Life Cycle

- Collect NFRS
- Analyze NFRs
- Input to
  - Design
  - Sizing
  - Business Flow
  - Zero Downtime System
  - WLM
- Root Cause Analysis & Suggestion



# Performance Engineering

## Testing vs. Engineering

Performance Testing	Performance Engineering
For quality check	For designing a tuned system
Verify - how a system will perform	Design - how a system will perform
Conduct once development is completed	Ongoing process through all phases of the development cycle
Knowledge of performance testing concept, tool operation, result analysis etc.	Knowledge of application design, architecture, development, tuning, performance optimization and bottleneck root cause investigation and fixing.
Identify the Performance Issues	Identify the root cause and resolve as per knowledge
Main focus on NFR, Test Result & Identified bottlenecks	Main focus on the performance of each component of the application
Follow Performance Test Life Cycle	Follow Performance Engineering Life Cycle

# Performance Testing

## Basics of Front-End testing

Alex Volkov 05/15/2022

# Front-End Performance Tests

## Google Vitals

- Frontend performance matters for accessibility, user experience, conversion rates, and of course, good old SEO (search engine optimization). To build fast websites, we need to know what to measure and how to measure it.
- Web Vitals is an initiative by Google to provide unified guidance for quality signals essential to delivering a great user experience on the web.
- Core Web Vitals are the subset of Google Web Vitals that apply to all web pages and they should be measured by all site owners.



# Core Web Vitals

The current set of Core Web Vitals metrics focuses on three aspects of the user experience—loading, interactivity, and visual stability—and includes the following metrics (and their respective thresholds):

- **Largest Contentful Paint (LCP)**: measures loading performance. To provide a good user experience, LCP should occur within 2.5 seconds of when the page first starts loading.
- **First Input Delay (FID)**: measures interactivity. To provide a good user experience, pages should have an FID of 100 milliseconds or less.
- **Cumulative Layout Shift (CLS)**: measures visual stability. To provide a good user experience, pages should maintain a CLS of 0.1. or less.

Web Vital	Fast	Moderate	Slow
First Contentful Paint	0 to 1.8 (sec)	1.8 to 3.0 (sec)	Over 3.0 (sec)
Speed Index	0 to 3.4 (sec)	3.4 to 5.8 (sec)	Over 5.8 (sec)
Time to Interactive (not used)	0 to 3.8 (sec)	3.8 to 7.3 (sec)	Over 7.3 (sec)
Total Blocking Time	0 to 200 (ms)	200 to 600 (ms)	Over 600 (ms)
Largest Contentful Paint	0 to 2.5 (sec)	2.5 to 4.0 (sec)	Over 4.0 (sec)
Cumulative Layout Shift	0 to 0.1 (unitless)	0.1 to 0.25 (unitless)	Over 0.25 (unitless)
Max Potential First Input Delay	0 to 130 (msec)	130 to 250 (msec)	Over 250 (msec)

# Example

## sitespeed.io

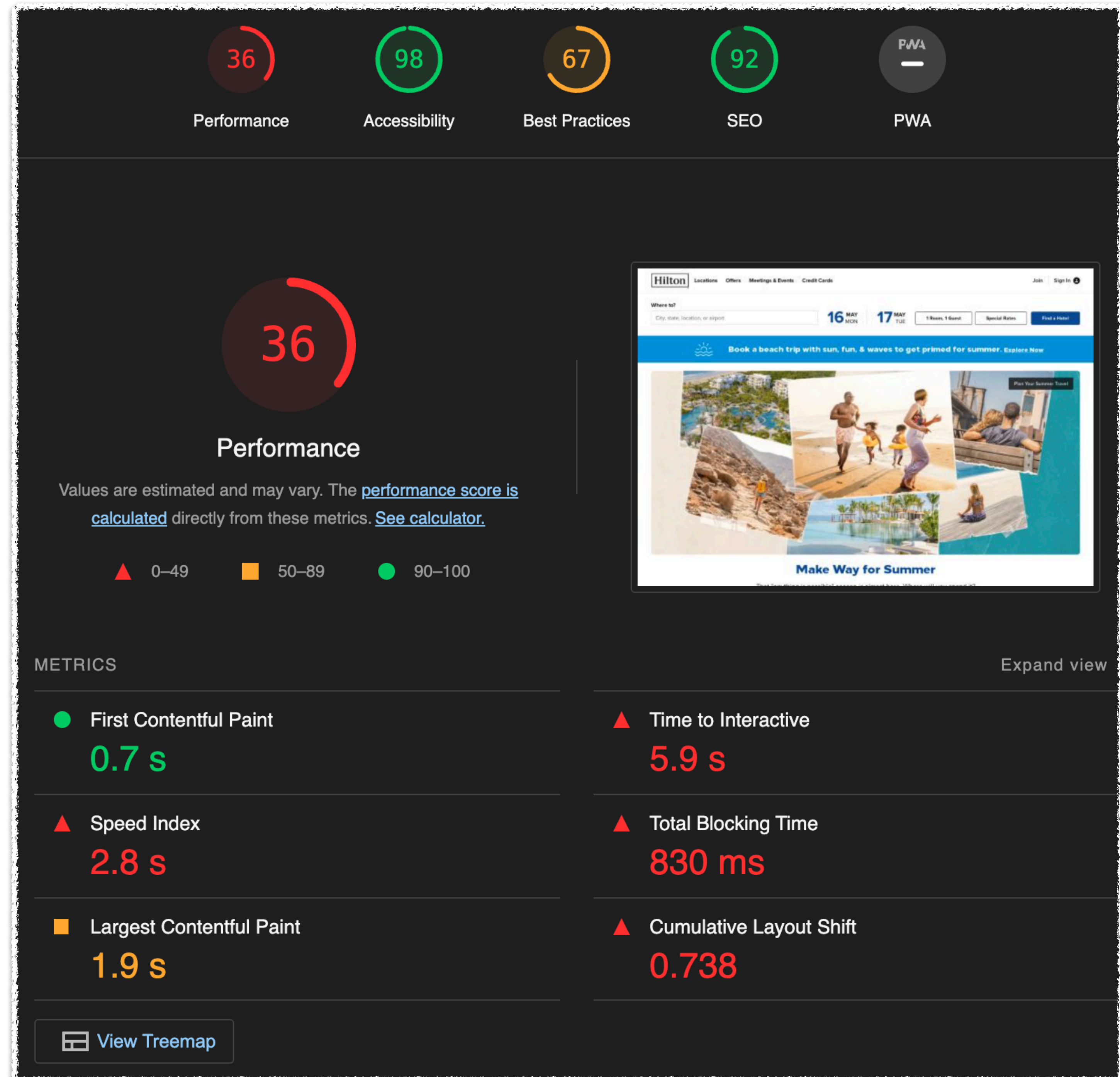
- New Tool
  - Uses Vitals
  - Automation
- Provides
  - Metrics
  - Suggestion
- Captures
  - Static
  - Dynamic

First Paint <b>1.723 s</b> (1.895 s)	First Contentful Paint <b>1.723 s</b> (1.895 s)	Fully Loaded <b>10.810 s</b> (11.699 s)
Page Load Time <b>4.846 s</b> (5.565 s)	Largest Contentful Paint <b>3.827 s</b> (4.133 s)	First Visual Change <b>1.103 s</b> (1.972 s)
Last Visual Change <b>7.455 s</b> (11.367 s)	Speed Index <b>3.533 s</b> (4.026 s)	Visual Readiness <b>6.352 s</b> (11.133 s)
TTFB <b>1.120 s</b> (1.242 s)	Cumulative Layout Shift <b>0.475</b> (0.879)	Total Duration of Long Tasks <b>1.564 s</b> (1.834 s)
Number of Long Tasks <b>13</b> (15)	Total Blocking Time <b>897 ms</b> (1.080 s)	Max Potential FID <b>473 ms</b> (510 ms)
Coach Overall Score <b>65</b> (66)	Coach Best Practice Score <b>77</b> (84)	Coach Privacy Score <b>72</b> (74)
Coach Performance Score <b>57</b> (60)	Total Requests <b>221</b> (268)	Image Requests <b>75</b> (81)



# Example Lighthouse

- Extension
  - Uses Vitals
  - Not Automation
- Metrics
  - Performance
  - Accessibility
  - Best practices
  - SEO
- Captures
  - Static

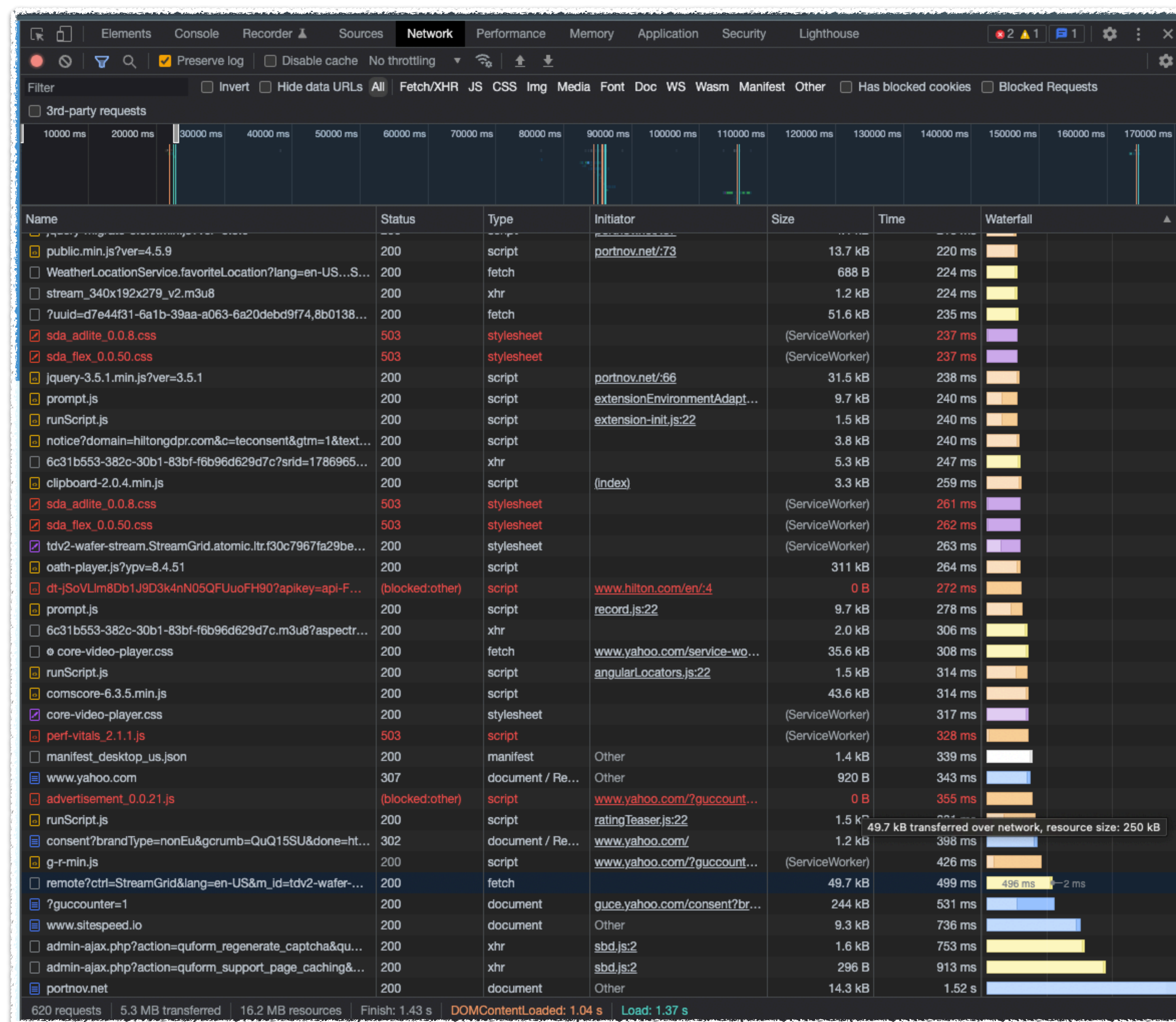




# Example

## Chrome DevTool

- Always available
- The view can be modified
- Not Specific



# Performance Testing

## Basics of Back-End testing

Alex Volkov 05/15/2022

# User Load

## Concurrent Users

- Virtual Load applied on the application
- Load Generator is used to generate the load
- Also called as Vuser, Thread, Virtual User etc.
- LG simulates the production user load on the performance test environment
- Type:
  - Average User Load
  - Peak User Load
  - Future User Load

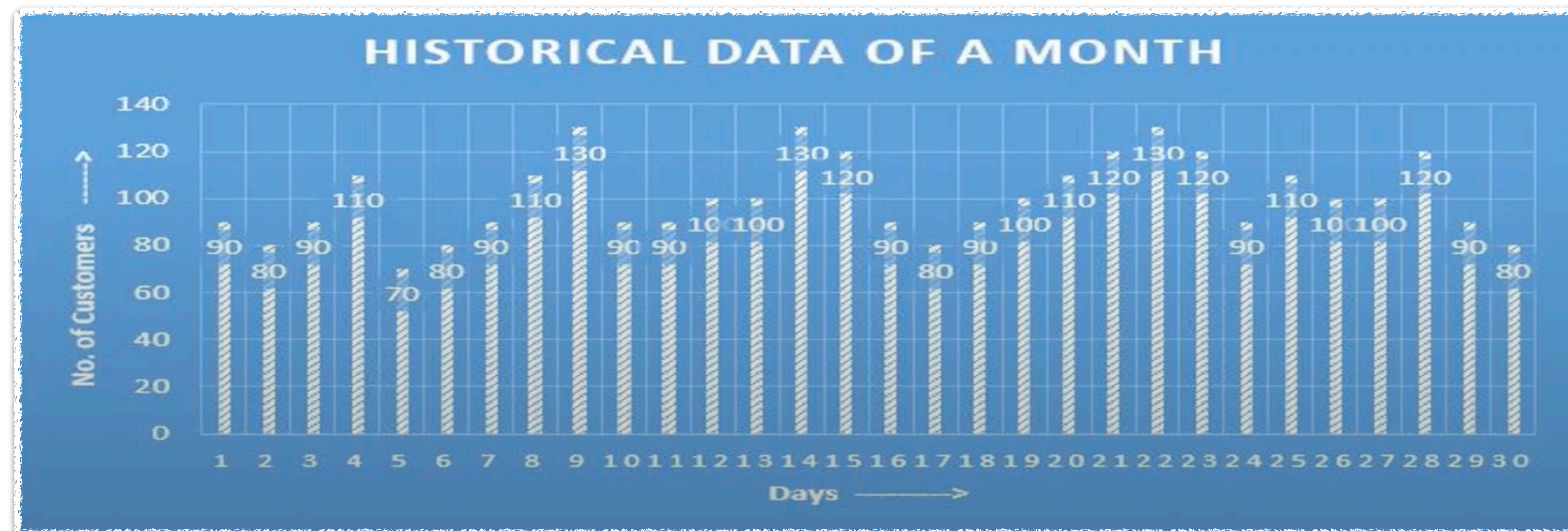


# Average user Load

## Historical Data

- Average of load identified during a day, a month or a year depends upon the production data selection criteria.
- Refer to the below graph shows the no. of customers visit in a month. The average of all the figures is 100.

**Note:** Can be obtained via Google Analytics or similar tools or from Data Analyst or DBA (Data Base Analyst)



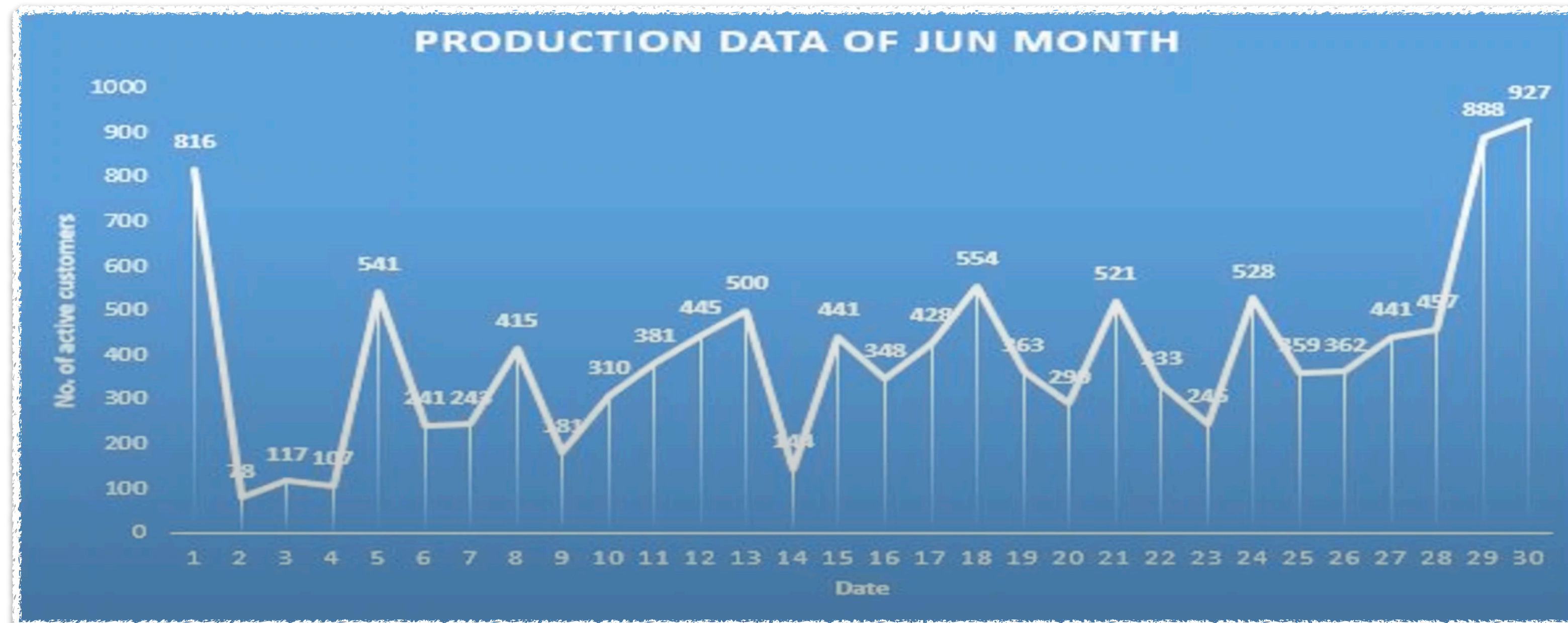


# Peak User Load

## Historical Data

- Peak Load is the highest load identified during a day, a month or a year depends upon the production data selection criteria.
- In the below graph, the highest number of active customers is 927 on 30th Jun. Hence 927 is the peak load for the load test.

**Note:** Can be obtained via Google Analytics or similar tools or from Data Analyst or DBA (Data Base Analyst)

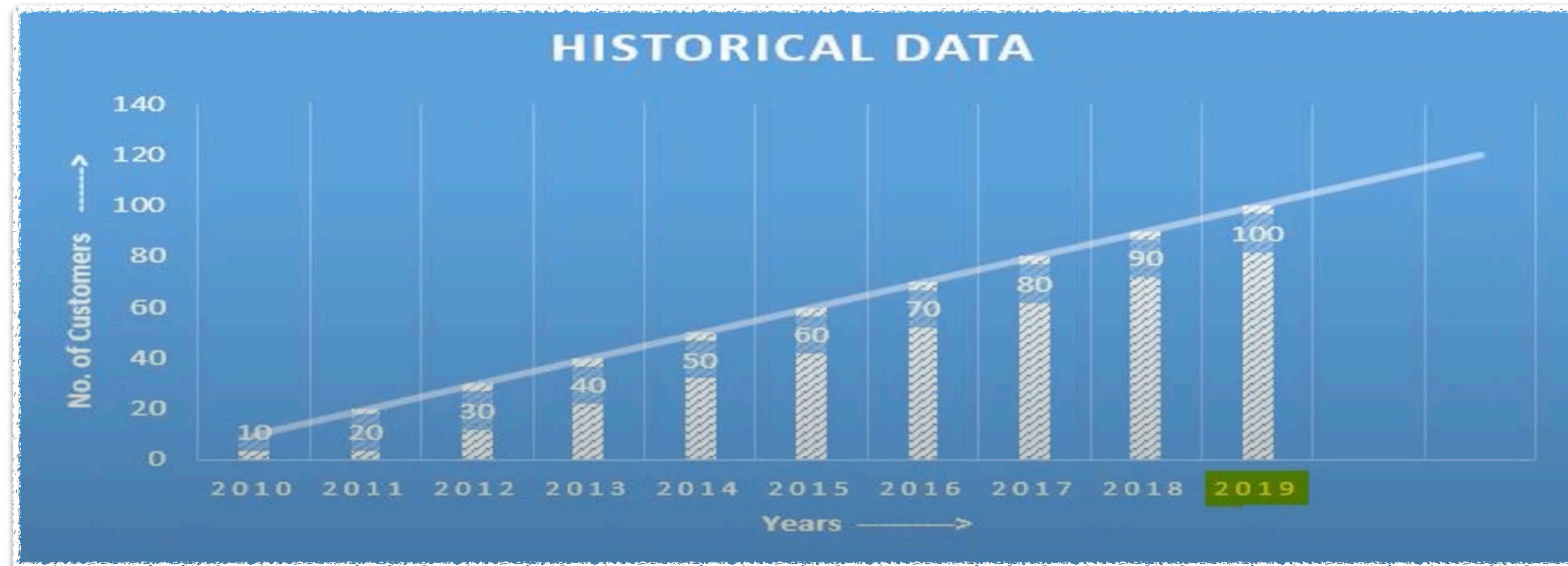


# Future User Load

## Estimation, usually is based on Historical data or Business Needs

- The future load is calculated with the help of historical data. BA predicts the numbers referring to previous years statistics or future business growth.
- Refer to the below graph shows the growth of no. of customers since 2010 till 2018. As per linear equation calculation, the predicted load is 100 for 2019. Hence 100 is the user load for the stress test.

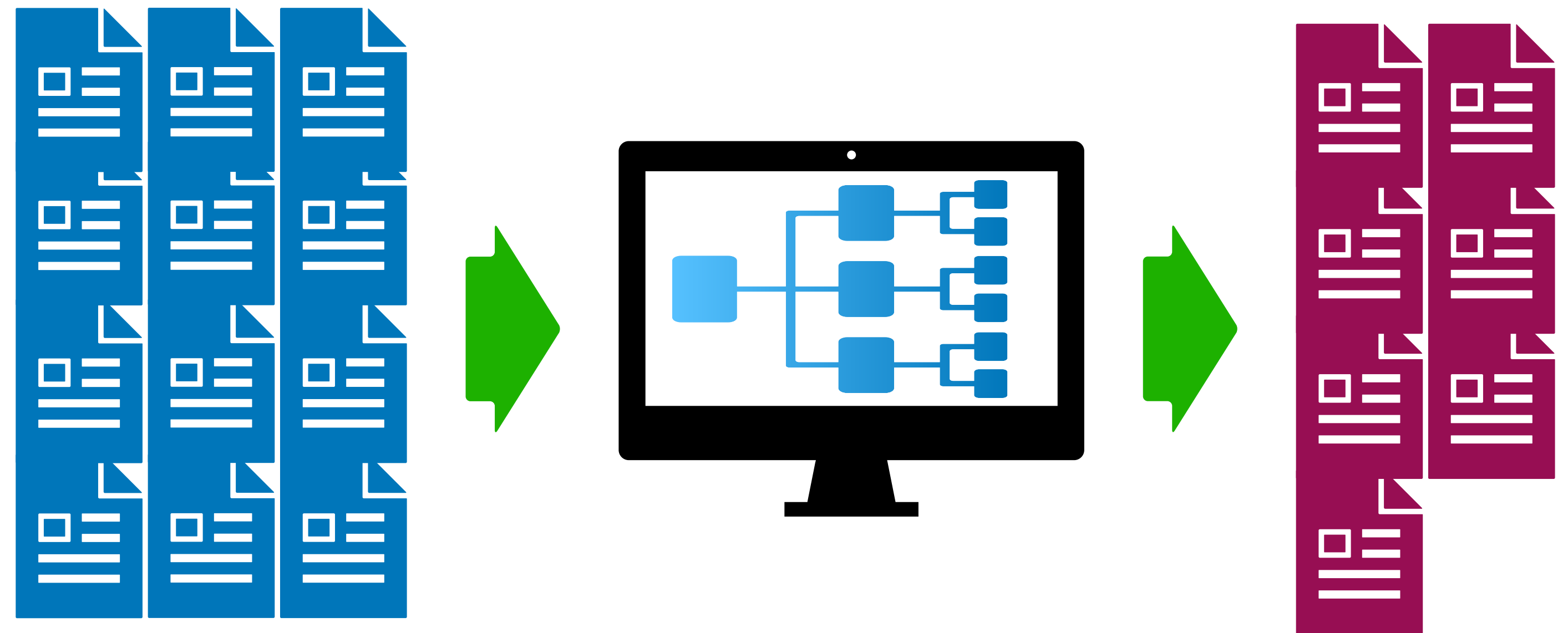
**Note:** Can be obtained from Business Analyst, Project Team or Product Owner



# Throughput

## General

- Throughput denotes 'rate'
- In Performance Testing, Throughput used for
  - Network
  - Server
  - Request
- Confusion?
- Example: Client NFRs are
  - User Load: 100
  - Response Time: 5
  - TPS: 0.5
  - Throughput: 200



# Performance Testing

## Metrics

Alex Volkov 05/15/2022



# Testing Metrics

- **Response Time**

- 90 Percentile
- 95 Percentile
- 99 Percentile
- Min, Max

- **Throughput**

- **Error %**

- **Resource Utilization**

- CPU, Memory, Disk, Connection, Latency, Thread, Heap Memory and etc.

Aggregate Report

Name:Aggregate Report

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only: ☐ Errors ☐ Successes

Configure

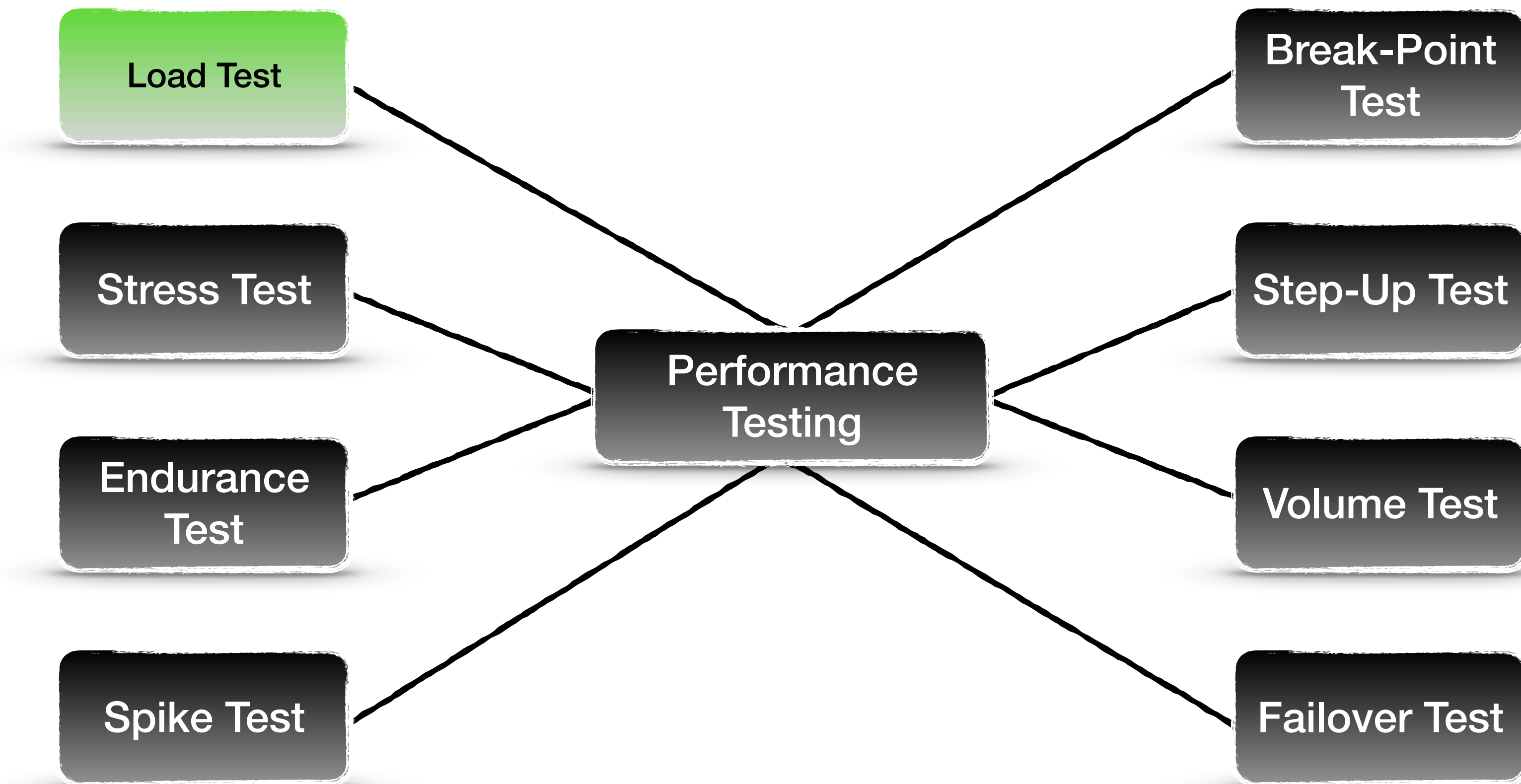
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB...	Sent KB/sec
homepage	1	219	219	219	219	219	219	219	100.00%	4.6/sec	265.74	45.28
Home page	1	219	219	219	219	219	219	219	100.00%	4.5/sec	260.97	44.47
signInForm	1	29	29	29	29	29	29	29	0.00%	34.5/sec	51.72	19.60
Login page	1	29	29	29	29	29	29	29	0.00%	1.6/sec	2.46	0.93
viewCatalog	1	150	150	150	150	150	150	150	0.00%	6.7/sec	22.53	71.79
signInAccount	1	237	237	237	237	237	237	237	0.00%	4.2/sec	15.33	54.68
Signin	1	387	387	387	387	387	387	387	0.00%	40.2/min	4.70	15.90
viewCategory	1	168	168	168	168	168	168	168	0.00%	6.0/sec	15.13	37.80
ViewCategory	1	168	168	168	168	168	168	168	0.00%	15.0/min	0.63	1.59
viewProduct	1	177	177	177	177	177	177	177	0.00%	5.6/sec	14.59	37.54
ViewProduct	1	177	177	177	177	177	177	177	0.00%	11.9/min	0.51	1.31
addItemToC...	1	161	161	161	161	161	161	161	0.00%	6.2/sec	17.17	41.39
newOrderFo...	1	172	172	172	172	172	172	172	0.00%	5.8/sec	17.18	38.63
setBillingInfo	1	169	169	169	169	169	169	169	0.00%	5.9/sec	15.94	47.35
TOTAL	14	175	169	237	237	387	29	387	14.29%	39.7/min	7.19	5.47

# Performance Testing

## Load Test

Alex Volkov 05/15/2022

# Types of Performance Testing



# Load

- The Load is the number of users that are using the application / webpage at the same time.
- Two types of Load
  - Main Load or just Load - the number of users that are completing the main end-to-end flows
    - Exmp: buying the item in the e-store
  - Bounce Load - the number of users that are using the application or web page different than main flows or incomplete flows
    - Exmp: just looking for that item in the e-store but never completing the transaction

# Load Test

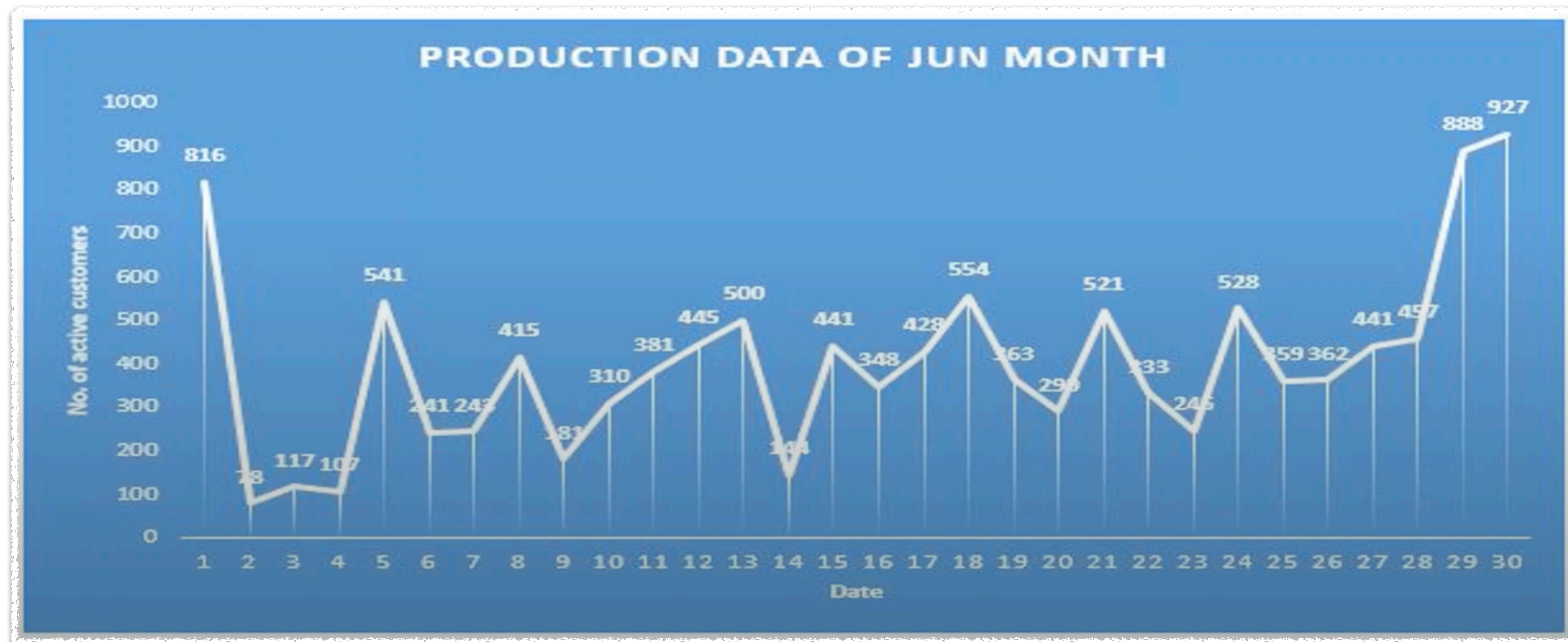
- Load Test verifies the performance of an application or system under a peak load condition.
- Load Test also validates the resource usage, stability and reliability of the software system under a peak load.
- Load Test duration could be in between ONE to TWO hours.
  - Common duration is ONE hour
  - Preferable TWO hours
  - Breaks 45 to 60 mins
- Common bottlenecks:
  - Unable to handle Peak user load,
  - Slow response time,
  - Exceed resource utilization



# Peak Load

## Existing application - History Check

- Peak Load is the highest load identified during a day, a month or a year depends upon the production data selection criteria.
- In the below graph, the highest number of active customers is 927 on 30th Jun. Hence 927 is the peak load for the load test.



# Peak Load

## New application - Estimation

- No production data to calculate the Peak Load
- Expected/ Estimated Peak Load value should be given by the client or business
- Estimation of Peak Load (200% or more of average load)
  - Use that estimation if no data is available at all
- Review the available market information of the closest competitors

# Approach of the Load Test

- There should be separate NFRs for Load Test
  - NFR - Non-Functional Requirements
- NFRs should have defined peak load, expected response time and TPS to be achieved
  - TPS - throughput per second. The amount of requests per second
- Design the workload model according to the given NFRs
  - The Workload is a model that describes the Loads, NFRs and TPS
- Add ramp-up and ramp-down period
- Monitor User Load, TPS, Response Time, CPU Utilization, Memory Utilization during the test
- Note down the identified bottleneck, error rate and timings for the investigation purpose



# Work Load Model

## Load Test

- The load test graph has peak load of 100 users. The total test duration is 1 hour and 20 minutes which includes 1 hour steady state and 10 minutes ramp-up and ramp-down period each.

1. Steady state duration usually as per NFRs.
2. The Load test doesn't require long steady states

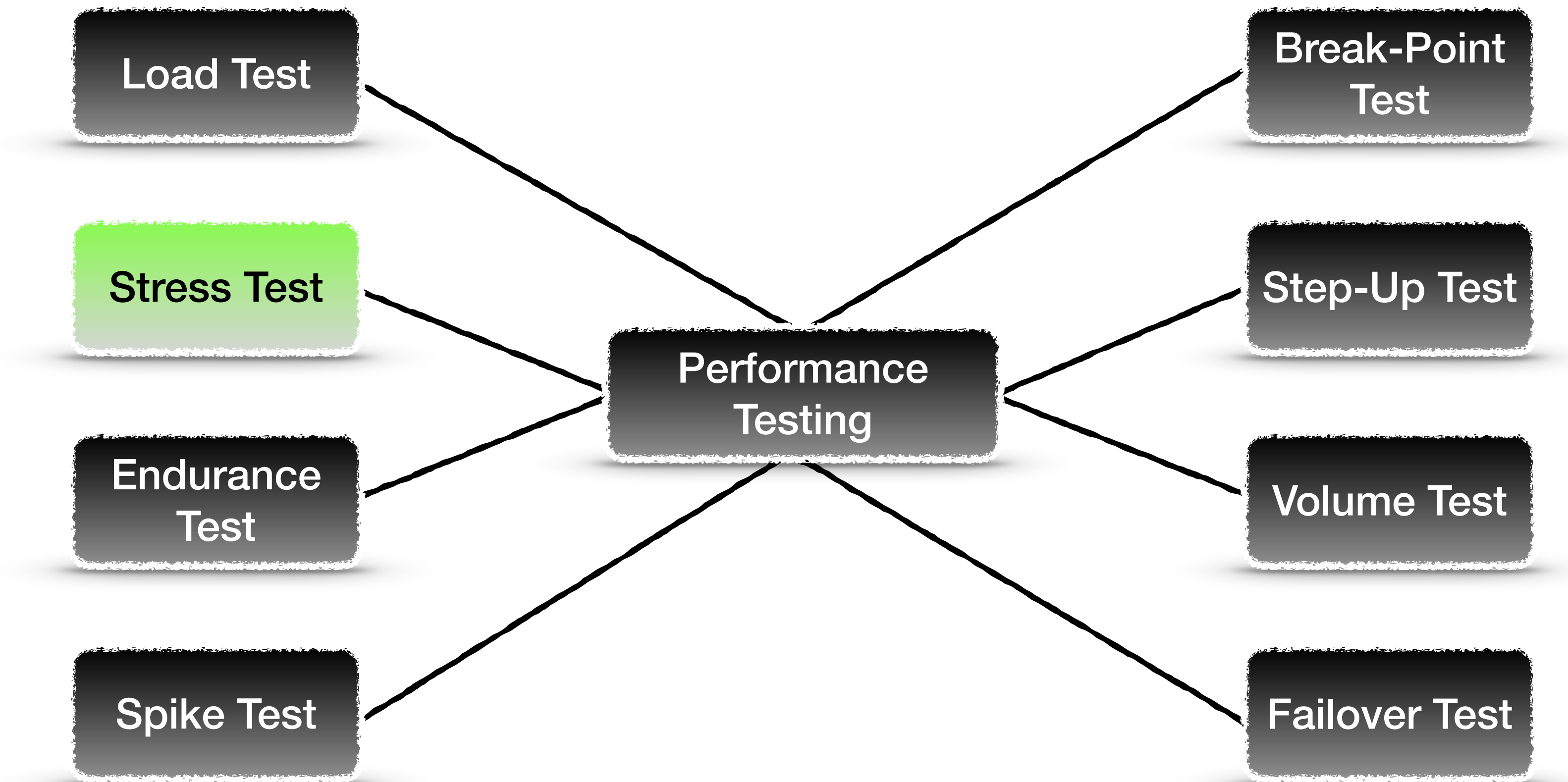


# Performance Testing

## Stress Test

Alex Volkov 05/15/2022

# Types of Performance Testing



# Stress Test

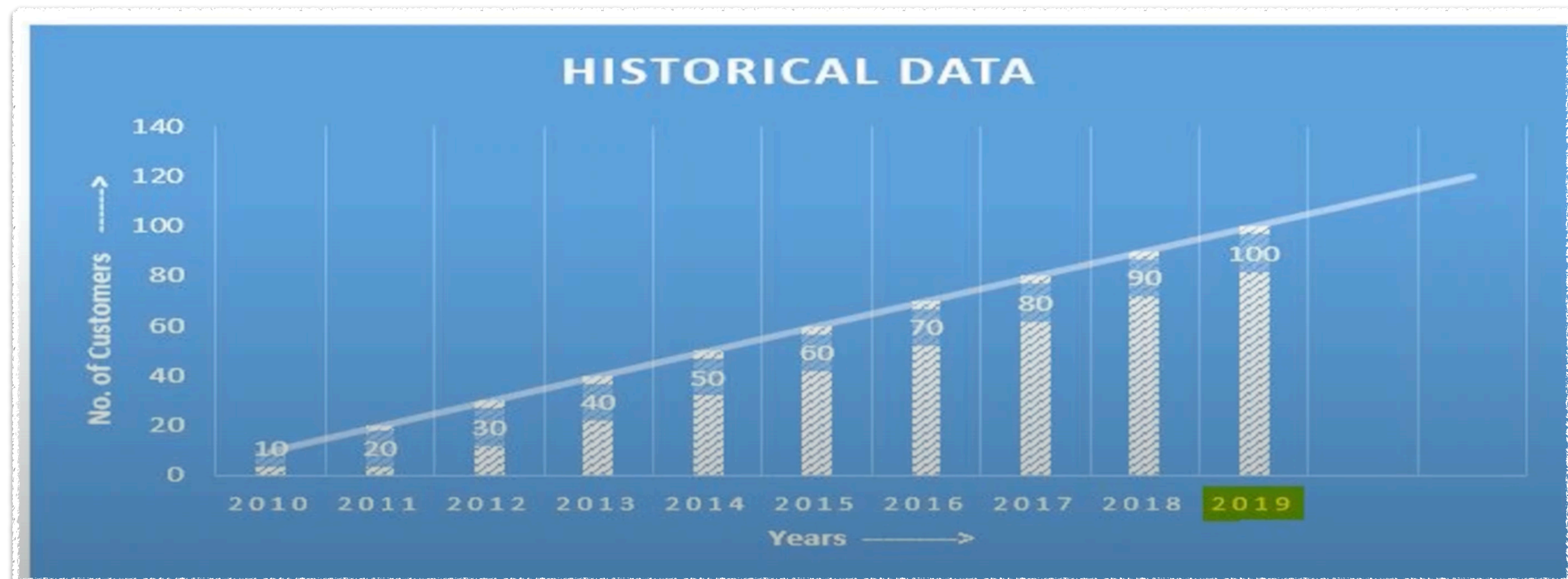
- Stress Test verifies the performance of an application or system under a futuristic load condition.
- Stress Test also validates the resource usage, stability and reliability of the software system under future load.
- Stress Test duration could be in between 1 to 2 hours.
- Common bottlenecks: Unable to handle provided user load, Slow response time, Exceed resource utilization, Application Crash during the test etc.



# Future Load

## Estimated Load

- The future load is calculated with the help of historical data. BA predicts the numbers based on the several years statistics and future business growth.
- Refer to the below graph shows the growth of no. of customers since 2010 till 2018. As per linear equation calculation, the predicted load is 100 for 2019. Hence 100 is the user load for the stress test.





# Future Load

## New Application

- No production data to calculate the future load
- Expected Future Load value should be given by the client or business as per the expected growth
- Prediction of Future Load (150% to 500% of the average load)
- Apply step-up load test approach

# Stress Test approach

- There should be separate NFRs for Stress Test
- NFRs should have defined future load, expected response time and TPS to be achieved
- Design the workload model according to the given NFRs
- Add ramp-up and ramp-down period
- Steady state duration usually as per NFRs
- Monitor User Load, TPS, Response Time, CPU Utilization, Memory Utilization during the test
- Note down the identified bottleneck, error rate and timings for the investigation purpose.

# Work Load Model

## Stress Test

- The stress test graph has peak load of 100 users. The total test duration is 1 hour and 20 minutes which includes 1 hour steady state and 10 minutes ramp-up and ramp-down period each.



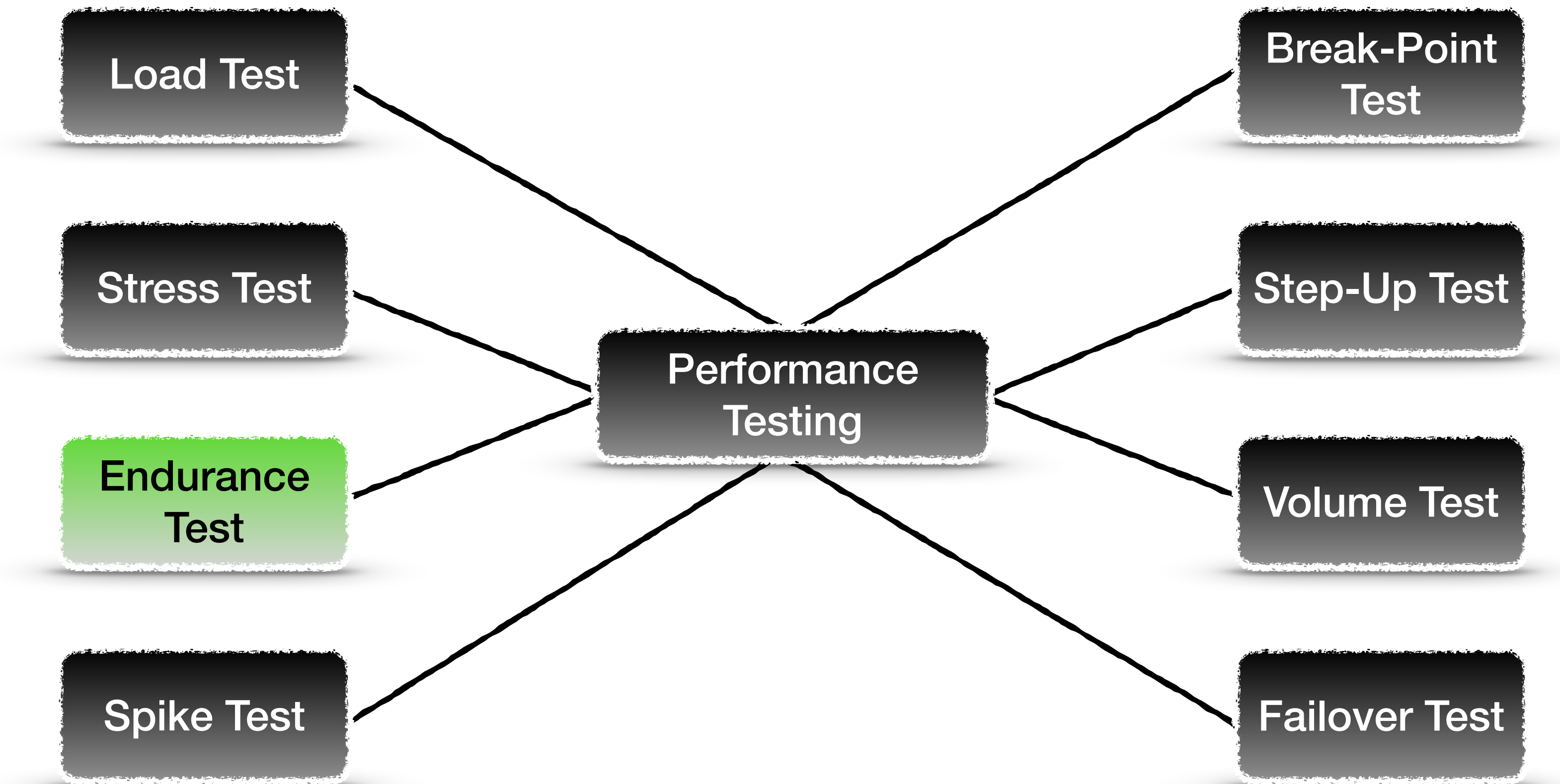
# Performance Testing

## Endurance Test

Alex Volkov 05/15/2022



# Types of Performance Testing



# Endurance Test

## Soak Test

- It is also known as a Soak Test.
- The Endurance Test verifies the performance of an application or system under continuous average load condition.
- The Endurance Test also validates the resource usage, sustainability and reliability of the software system under average load. The main purpose of Endurance Test is to find out the memory (resource utilization) leakage.
- It is a longer duration test typically takes from 8 to 24 hours.
  - If observed results aren't trending/benchmarking with other tests it could be repeated
- Common bottlenecks: Memory leakage, Slow response time after a certain period of time, Excessive Resource Utilization in between the test, Application Crash during the test etc.

# Memory Leakage

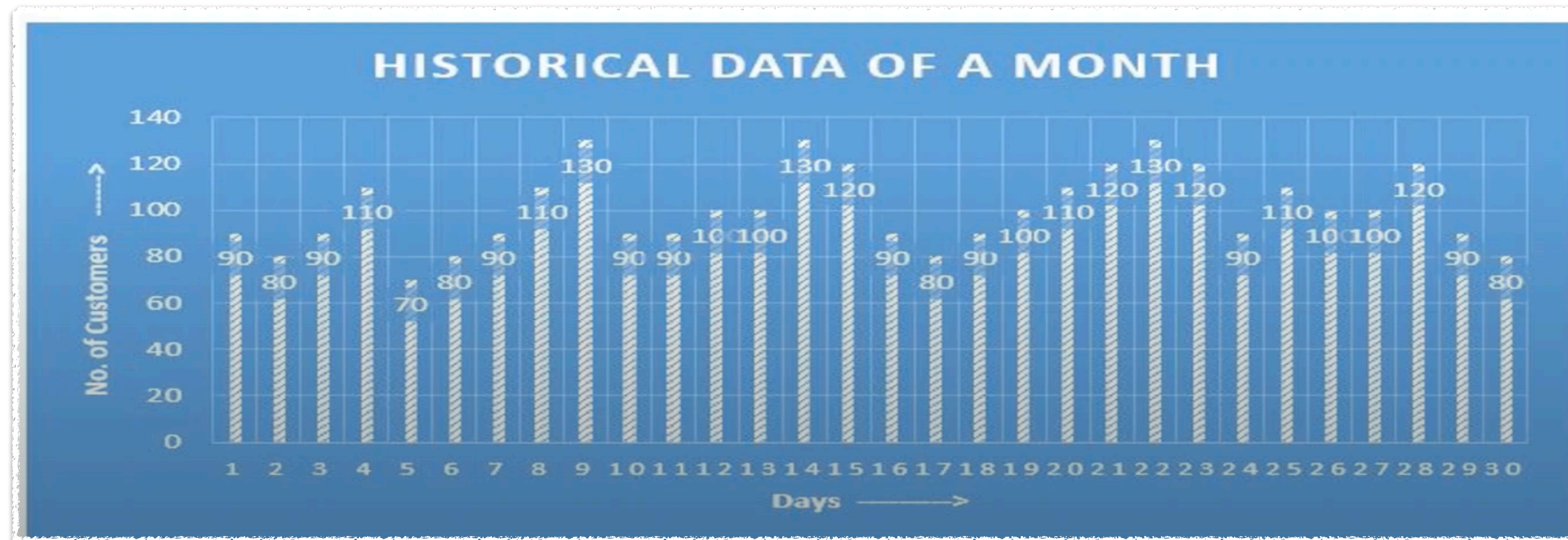
- When a computer program consumes memory but is unable to release it back to the operating system.
- Symptom:
  - Gradually increase in the memory usage graph
  - Memory utilization remains the same even after the completion of the test
- Causes:
  - Programming errors (code errors)
  - Improper Garbage Collection
  - Deadlock
    - Code related, when programming object can't be executed because it has not enough resources



# Work Load Model

## Endurance Testing - Historical Data

- Average of Load identified during a day, a month or a year depends upon the production data selection criteria
- Refer to the below graph shows the number of customers visit in a month. The average of all the figures is 100. Hence 100 is the user load for the Endurance Test.





# Work Load Model

## New Application

- No production data to calculate the average load
- Expected average load value should be given by the Client or business as per the expected growth.
- Apply step-up load test approach
  - Based on the assumption start to increase the number of concurrent users
  - The Load of similar applications can be taken into account

# Endurance Test Approach

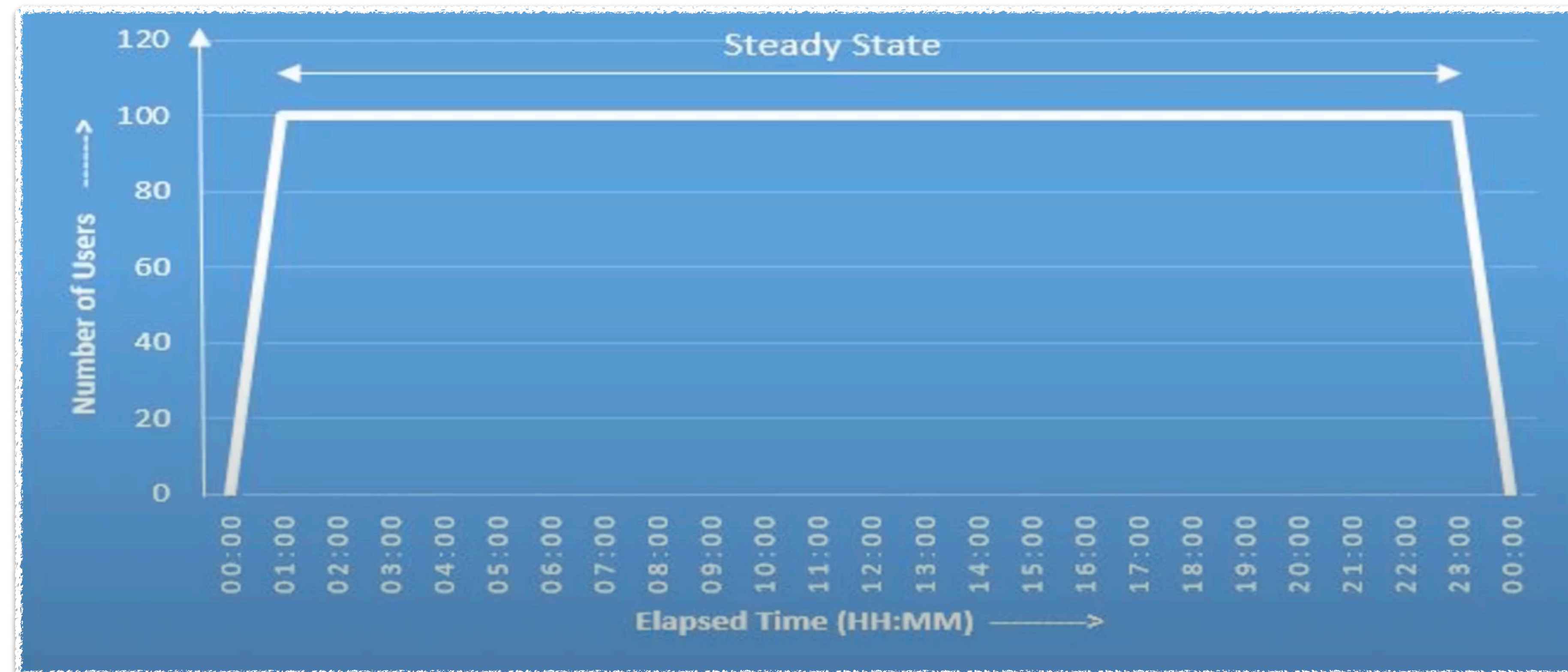
## Soak test

- There should be separate NFRs for Soak Test NFRs should have defined user load, expected response time, TPS to be achieved and error rate
- Design the workload model according to the given NFRs
- Add ramp-up and ramp-down period
- Steady state duration should be as per NFRs (minimum 8 hours)
- Monitor User Load, TPS, Response Time, CPU Utilization, Memory Utilization during the test
- Note down the identified bottleneck, error rate and timings for the investigation purpose.

# Work Load Model

## Endurance test

- The Soak test graph has peak load of 100 users. The total test duration is 24 hours which also includes the ramp-up and ramp down period



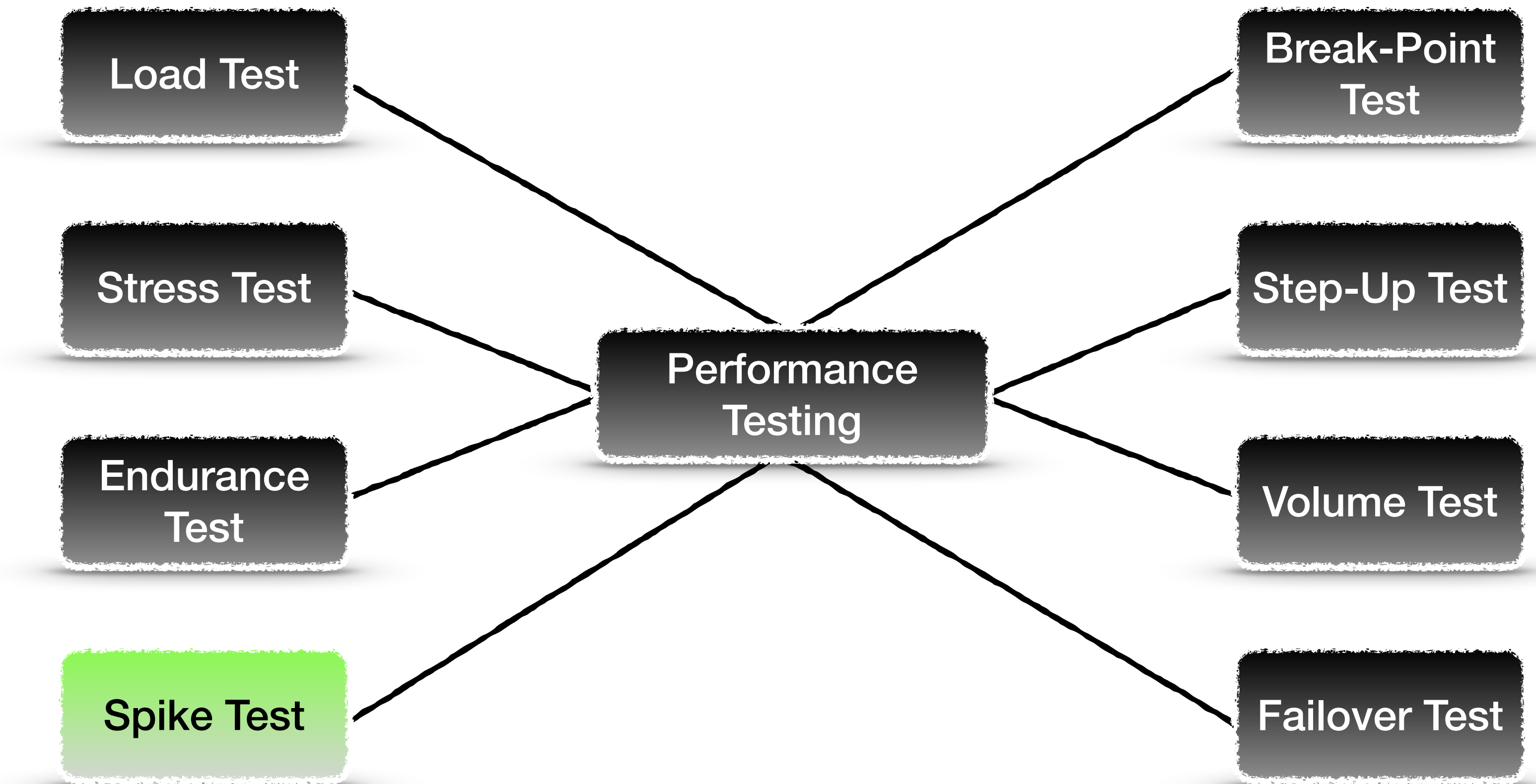
# Performance Testing

## Endurance Test

Alex Volkov 05/15/2022



# Types of Performance Testing



# Spike Test

- Spike Test verifies the performance of an application or system under the condition when load suddenly increases for a short duration.
- It does not fall under core performance testing.
  - It is a type of an additional test.
- Duration: Typically 1 hour (Spike Duration: 1 to 5 minutes)
- Common bottlenecks:
  - Application Crash at the time of spike, application recovery difficulties and etc.

# Purpose

## Spike Test

- Verify the sustainability of the application at the sudden load increase (overload)
- Identify the deviation in the response time during spike load
- Check the failure percentage of the transactions
- Identify the type of error like 5XX
  - Note the recovery time of the application in case of application is down during spike
- Check whether the Resource Utilization as CPU, Memory and Disk do not breach the defined performance limit even during the spike period

# Work Load Models

## Spike Test - Historical data

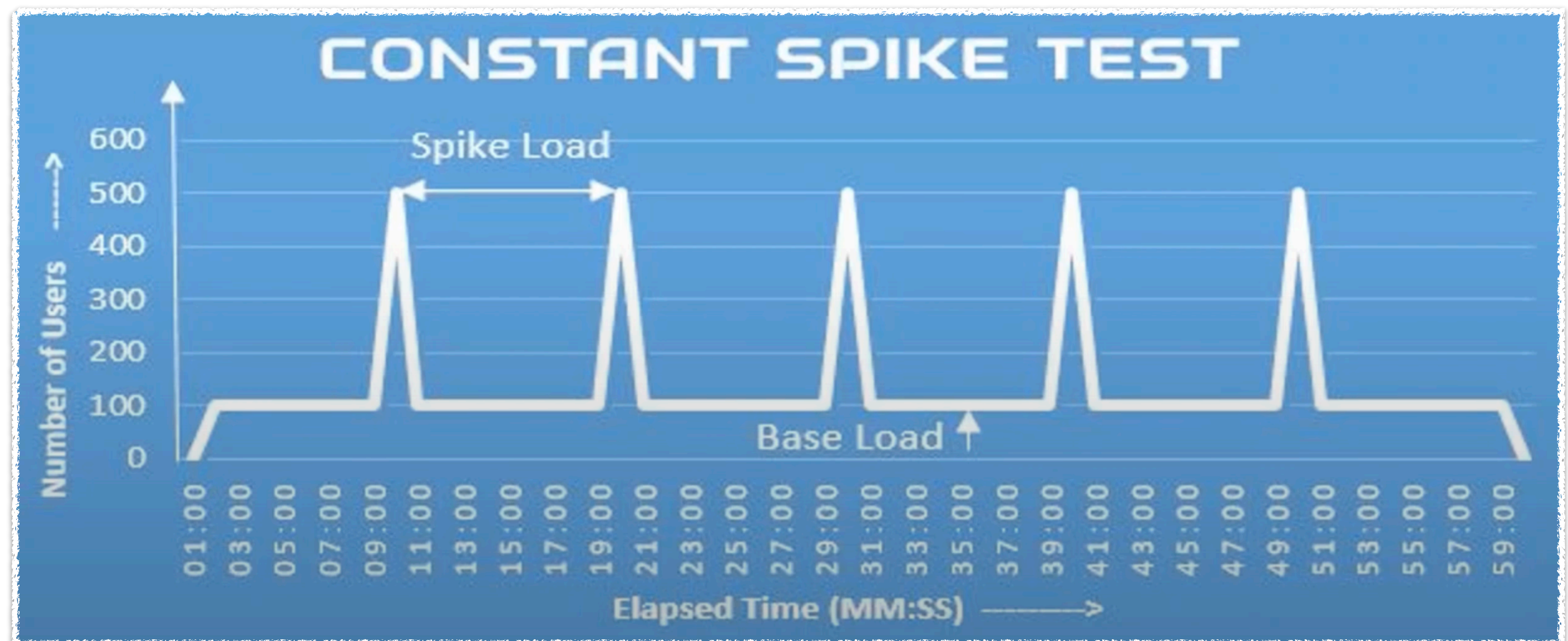
- There are 3 types of workload model of Spike Test:
  - Constant Spike Workload Model
  - Step-up Spike Workload Model
  - Random Spike Workload Model
- Required load type to design the workload for spike test
  - Spike Load
  - Base Load



# Work Load Model

## Constant Spike Test

- A constant spike load applies to the server after a certain interval of time. In this type of spike test, all the spike have same height i.e. the same spike load

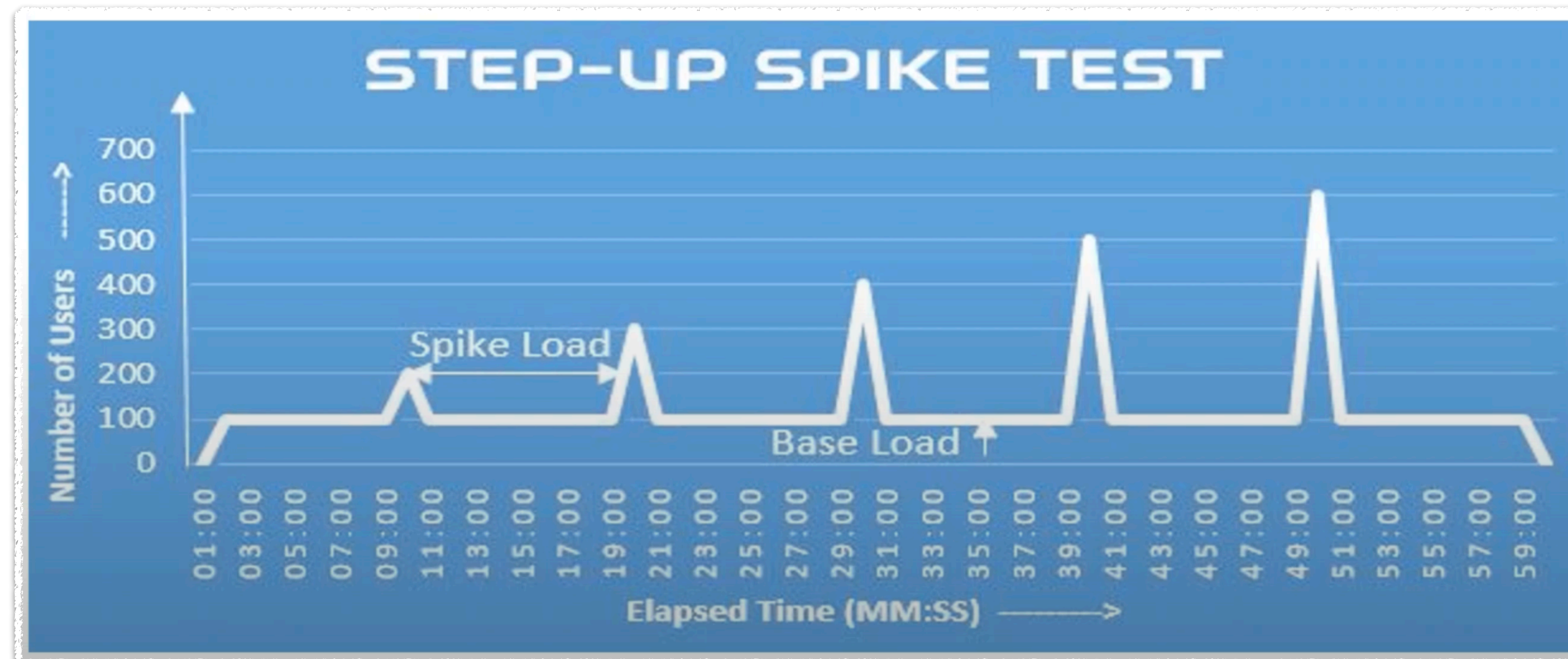




# Work Load Model

## Step-Up Spike Test

- A gradually increase spike load applies to the server after a certain interval of time.
- Response time should measure at each spike and analyze how much it deviates from base load response time?

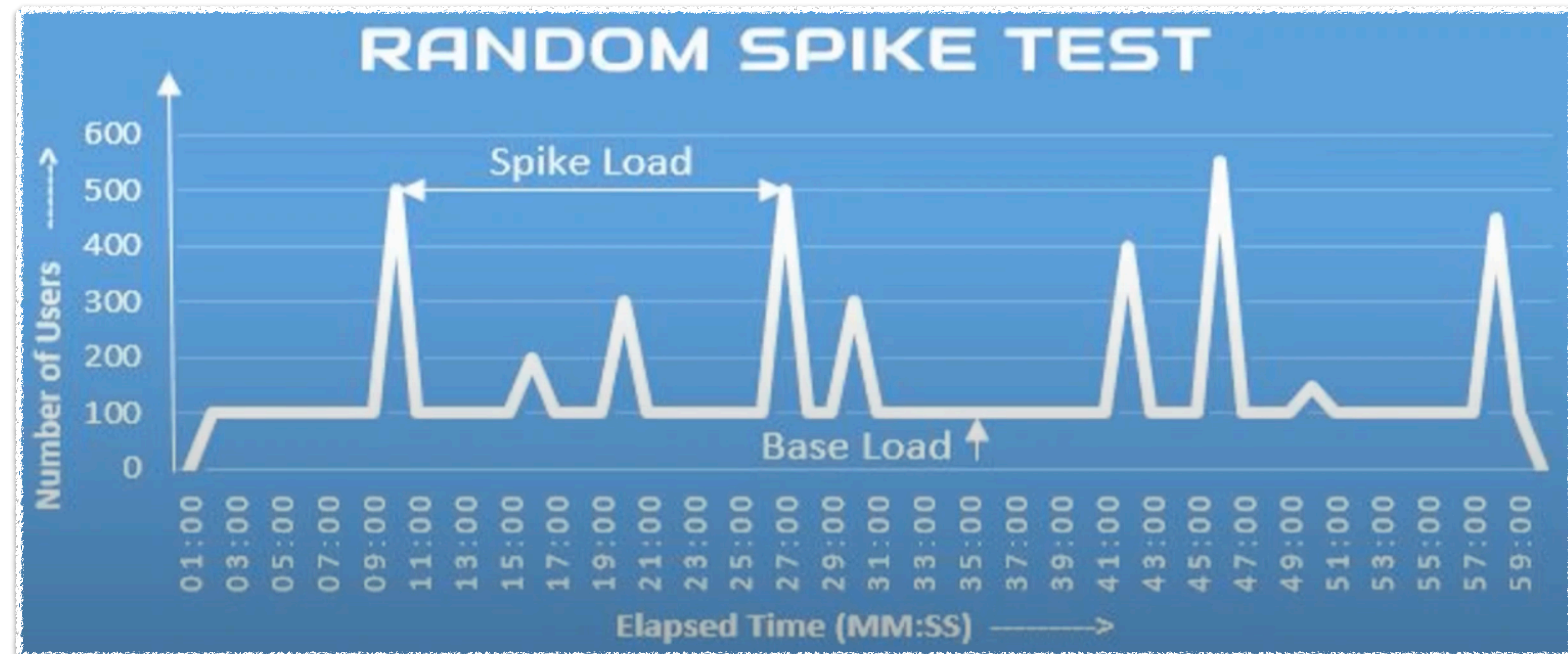




# Work Load Model

## Random Spike Test

- A random spike load applies to the server at random interval. Such a test is conducted for an application that frequently gets spikes in the production environment



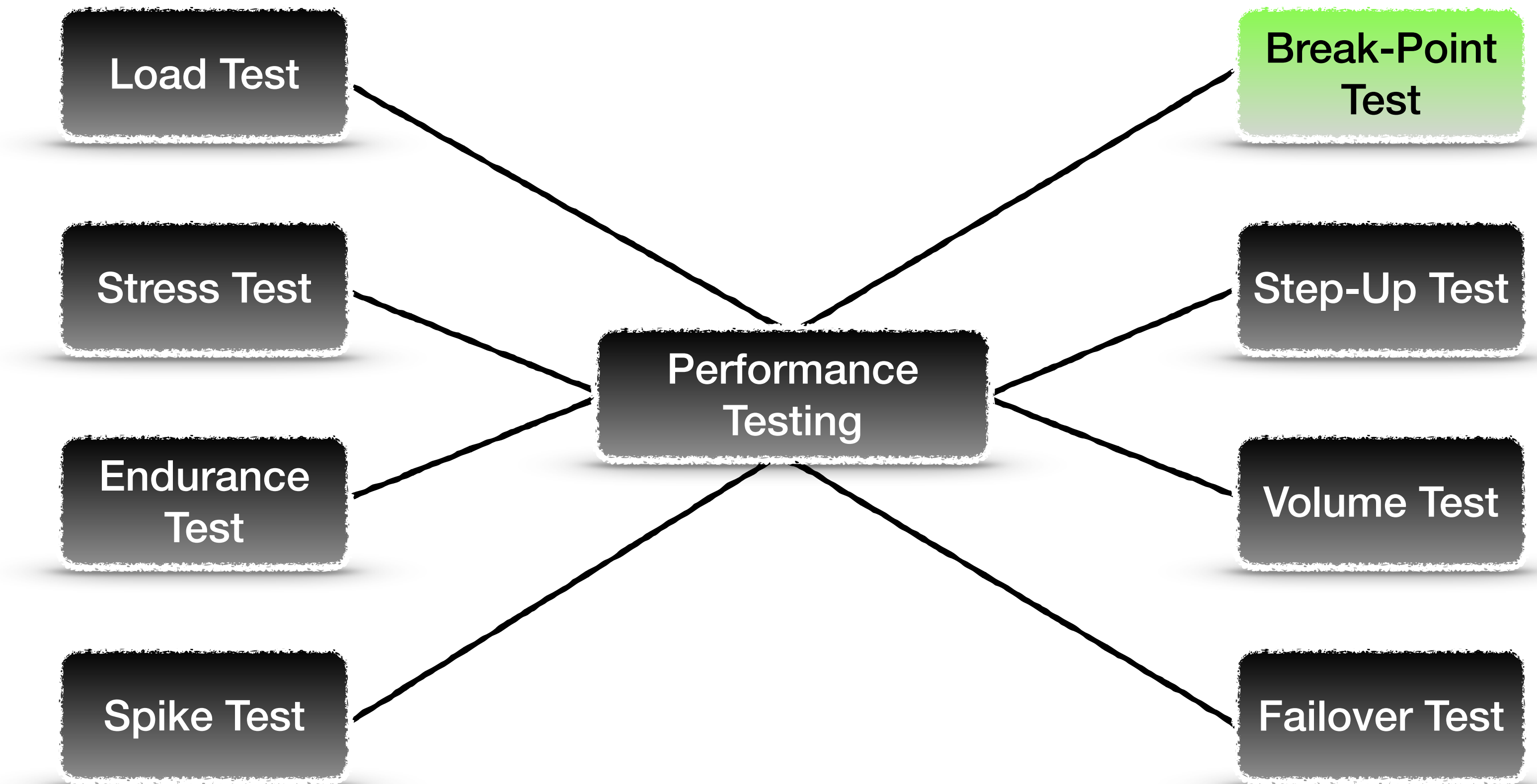
# Performance Testing

## Break-Point Test

Alex Volkov 05/15/2022



# Types of Performance Testing



# Break-Point Test

## Break-Down Test

- It is also called as Break-Down Test, Application Failure Test, Fatigue Test etc.
- Negative Testing Feature - Break the application is a systematic way
  - Break-Point Test is conducted to identify the user load at which the application gets failed.
  - Results provide the maximum user handling capacity of the application
- Break-point in terms of TPS rate
- Note down the resource utilization at break-point
- Investigate the causes of application failure
- If application does not meet the minimum acceptance criteria, then it requires tuning.

# Purpose of Break-Point Test

- Identify the critical breaking load for the application
- Check the response time behavior of the application at a different level of load
- Identify the behavior of the server resources
- Analyze which component fails first and why?
- Determine the impact of the failure on other systems
- Investigate the more tuning opportunities in the application

# Common causes of Failure

- Software Failure
  - Programming errors
  - Improper Garbage Collection
  - Deadlock
    - Code related, when programming object can't be executed because it has not enough resources
- Hardware Failure
  - Resource Capacity
    - CPU
    - Memory
    - Disk (I/O) - simply means that system cannot complete the read/write operation to the disk.

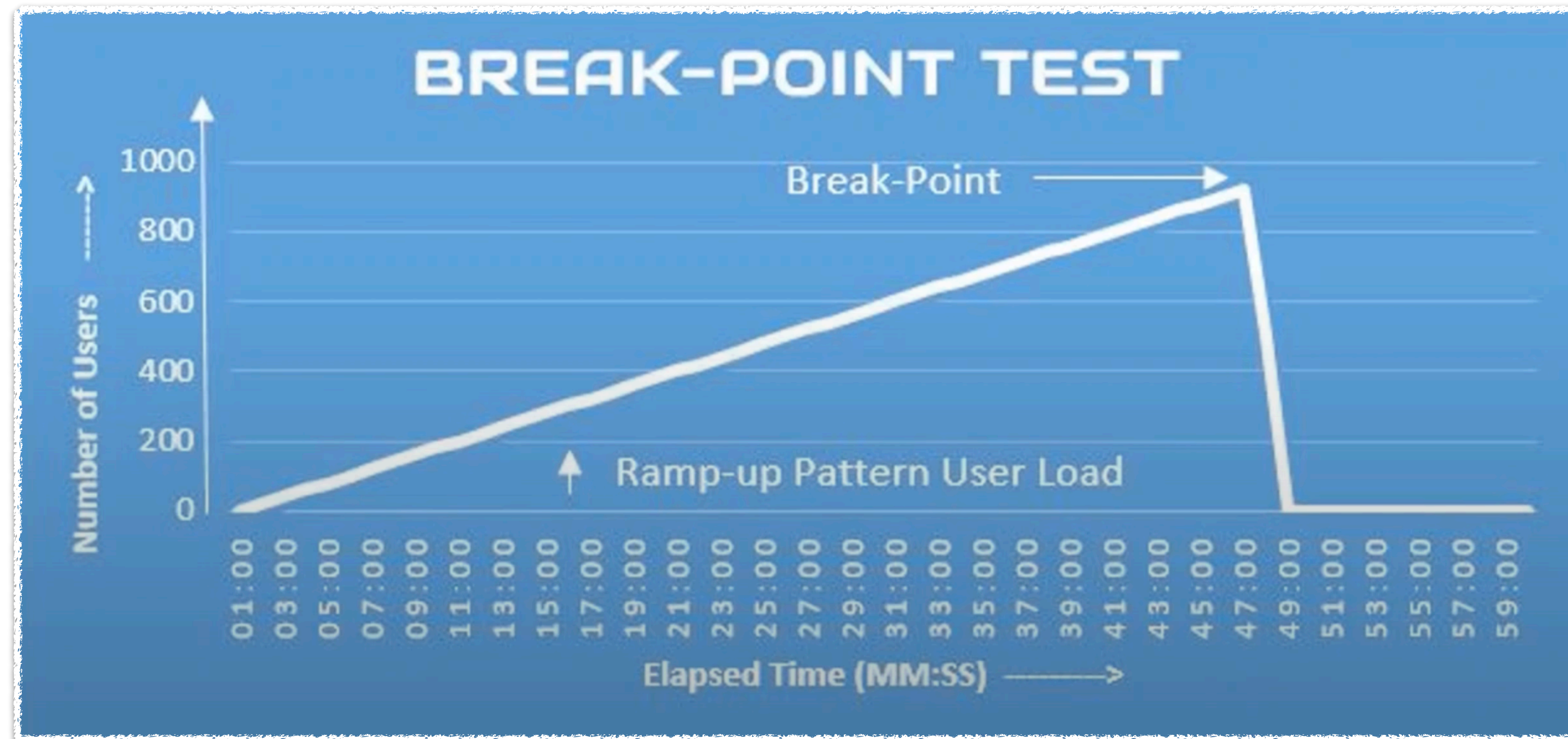


# Break-Point Test Approach

- Open-ended Requirement i.e. No Pass or Fail criteria
- No quantitative NFRs for user load, response time, TPS and error rate
- Incremental user load model is used as workload model
- Continue till application is fully down (500/Internal Server Error, 504/Service Unavailable etc.)
- Monitor User Load, TPS, Response Time, CPU Utilization, Memory Utilization during the test
- Note down all the important metrics reading at different user load for the investigation purpose.
- Conduct more than one tests to find out the correct break-point
- If the difference is more then it is advisable to conduct another test and average out the values of those tests which have the closest break-points.

# Work Load Model

- The main aim of this test is to break the system so the user load applies to the server is in ramp-up pattern. There is no steady state and hence a linear graph is formed

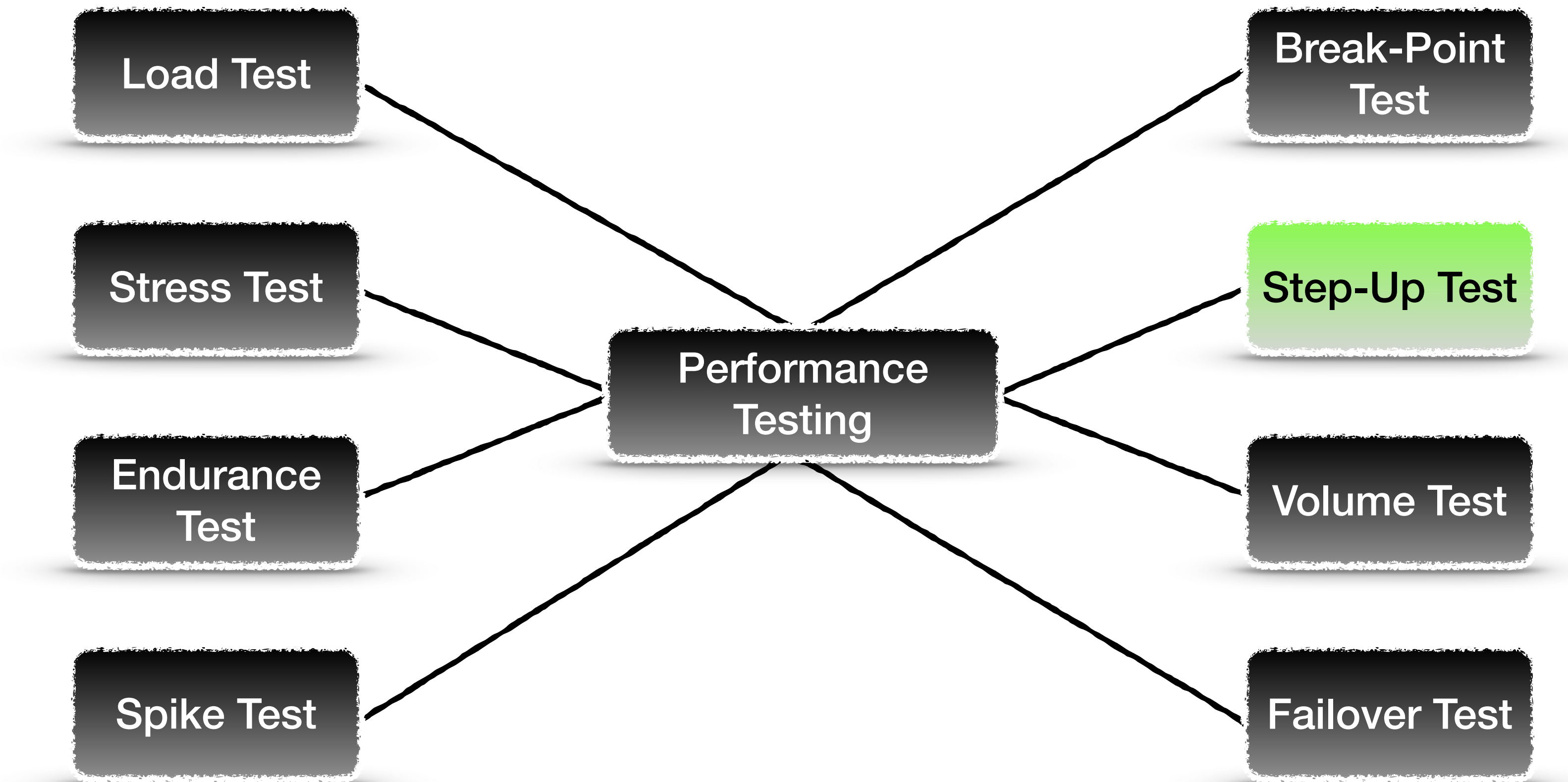


# Performance Testing

## Step-Up Test

Alex Volkov 05/15/2022

# Types of Performance Testing





# Step-Up Test

## Scalability Test

- Step-up Test verifies the performance of an application at varying load
- Multiple steady states of a certain duration
  - Also called as
  - 'Multi-level Load Test'
- Duration of the test depends on the number of steps and the duration of each step in the workload model.
- Common bottleneck:
  - Application may crash at higher load

# Purpose of Step-Up Test

- Identify the behavior of the application at a different level of load
- Identify the deviation in the response time during Step-up load
- Check the failure percentage of the transactions
- Check the performance of the new application where NFRs are not defined
- Conduct a hybrid performance test on the application
  - Load Test, Stress test and Break-Down Test

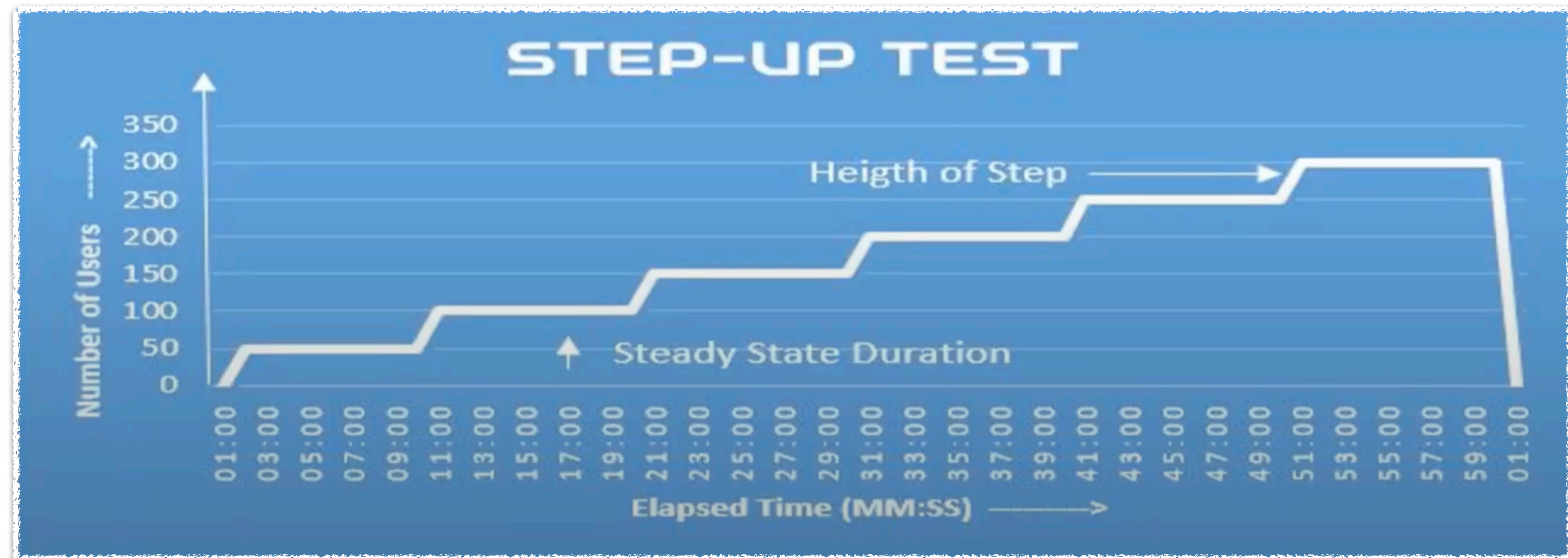
# Step-Up Test Approach

- NFRs (if any) should have defined user load, expected response time and error rate
- You can also use soak and stress test user load NFRs to design the step-up load scenario
- For a new application start the test with less user load and increase the load gradually
- Design the workload model according to the given NFRs
- Add ramp-up and ramp-down period
- Steady state duration of each step should be as per NFRs
- Monitor User Load, Response Time, CPU Utilization, Memory Utilization, error rate during the test
- Note down the identified bottleneck, error rate and timings for the investigation purpose

# Work Load Model

## Step-Up Test

- 3 required inputs:
  - Height of the step (User load)
  - No. of steps
  - Steady State Duration at each step



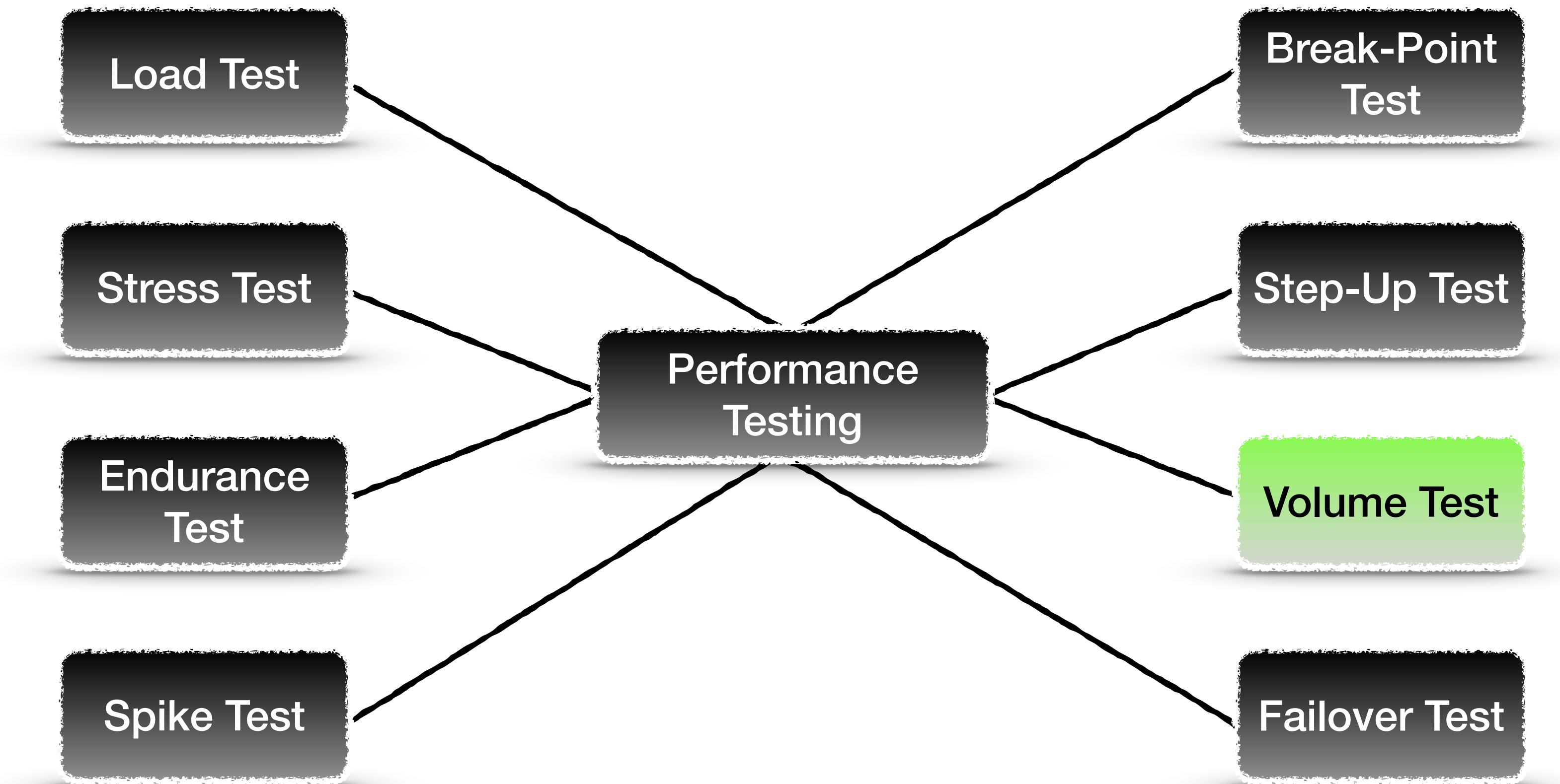


# Performance Testing

## Volume Test

Alex Volkov 05/15/2022

# Types of Performance Testing



# Volume Test

## Batch or Flood Test

- Volume Test verifies the performance of an application which deals with high volume. E.g. Banking and Telecom domain applications
- Also called as Batch Test or Flood Test
- In most of cases, it is performed manually
  - Developers can perform this type of test
- Automation of Test Data creation
- Duration of the test depends on the scenario. Mostly concentrate on batch completion timing
- Common bottleneck
  - Application Crash
  - Application fails to process the data

# Purpose of Volume Test

- To check the performance of a middleware system
- To check the performance of the Database
- To check the performance of the Messaging
- To identify the batch processing time
- To find-out the processing time of certain number of files/data
- To check the data loss issue
- To validate the resource utilization during batch processing



# Volume Test Approach

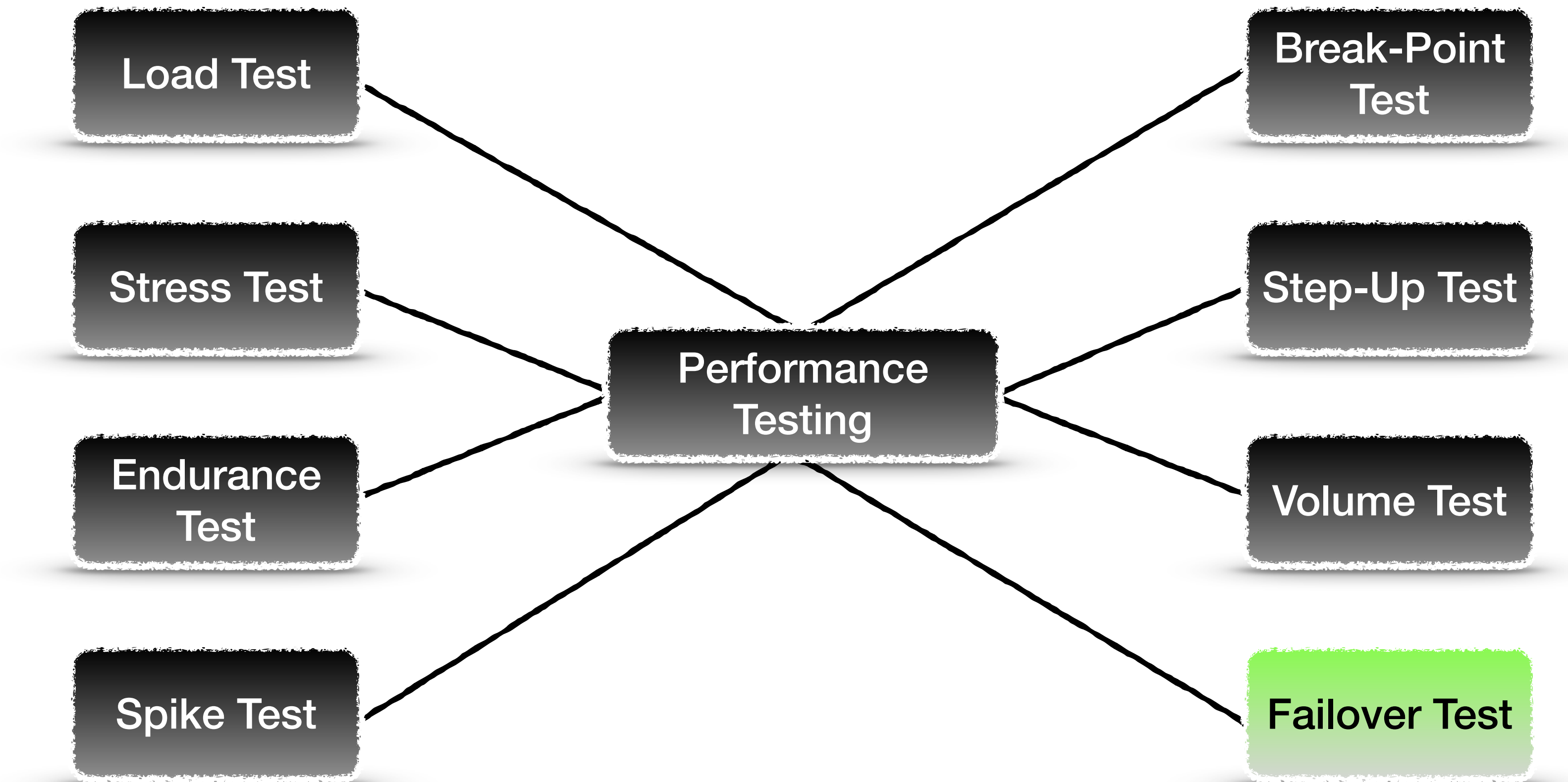
- Clarify and Understand the NFRs properly
- As per NFRS
  - Create the number of batches/files
  - Number of records in each batch/file
- Discuss with the project team and get the knowledge on 'How the batch/file will be processed'?
- Prepare the performance testing approach according to the method of processing of batch/file/data
- Capture the processing time and validate against the given NFRs
- Verify - there should not be any data loss or false processing
- Special attention should be given on resource utilization
- Raise the defect if bottleneck is found

# Performance Testing

## Failover Test

Alex Volkov 05/15/2022

# Types of Performance Testing



# Failover Test

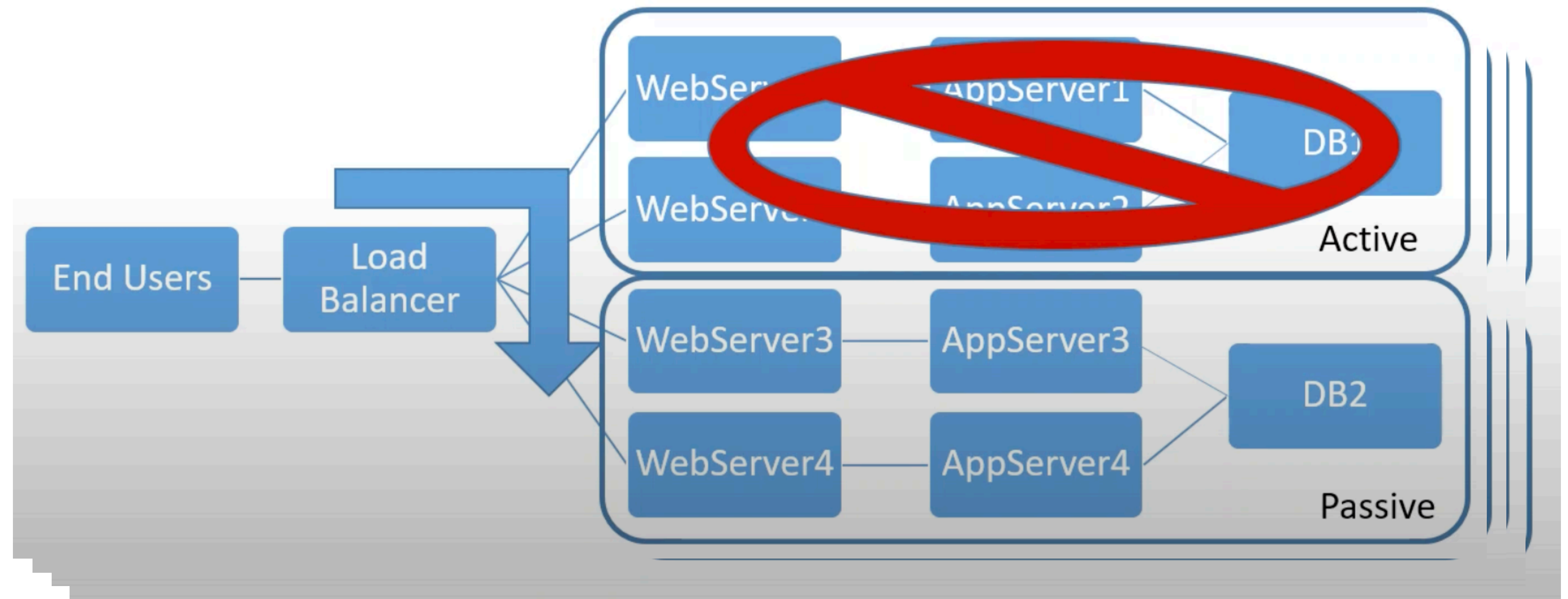
- Failover Test is different from other performance tests
- Other Performance Test:
  - To check the performance of the system till no failure
- Failover Test:
  - To check the performance of the system during the failure
- Types of failover test:
  - Software Failover Test
  - Hardware Failover Test
- Duration of the test depends on the expected system recovery time. Ideally 1 hour scenario is designed.
- Common bottlenecks:
  - Application fails to recover
  - End user gets error message until application is recovered
  - Data/Request Loss



# Failover Test

## Example

- A system has Four Web-Servers, Four App-Servers and Two DBs.
- There is one load balancer which distribute the load on both the instances equally



# Purpose of Failover Test

- To check the performance of the system in case of failure
- To validate the expected recovery time
- To identify the capacity of back-up system
- To test the working of Load Balancer
- To test the system which has zero downtime even during back-up, maintenance and recovery task
- To verify the disaster management system
- Impact of failure on the business

# Type of Failover Test

- Software Failover Test
  - To verify the behavior of the application when some of the components are failed due to software issue like Load balancer failure, GC issue, Queue Piled-up etc.
- Hardware Failover Test
  - To verify the behavior of the application when some of the components are failed due to hardware issue like Connectivity breakage, Database center failure due to natural calamities etc.

# Failover Test Approach

- Clarify the NFRs properly
- Discuss with the project team and get the knowledge on 'What is process to down the server leg?' and How to physically disconnect the server for hardware failover test?
- Design the failover scenario as per given NFRs by dividing the steady state into 3 sub-states
  - Pre-failover Period (Normal Steady State)
  - Failover Period
  - Recovery Period
- Start the test with user ramp, attempt the steady state, down the component, observe the behavior of the application and server statistics which are in up state, validate the auto-recovery of the application, continue the test for some time after application recovery then ramp-down the user.
- Verify - there should not be any data loss or false processing
- Special attention should be given to resource utilization
- Raise the defect if bottleneck is found



# Performance Testing

End of Module

Alex Volkov 05/15/2022