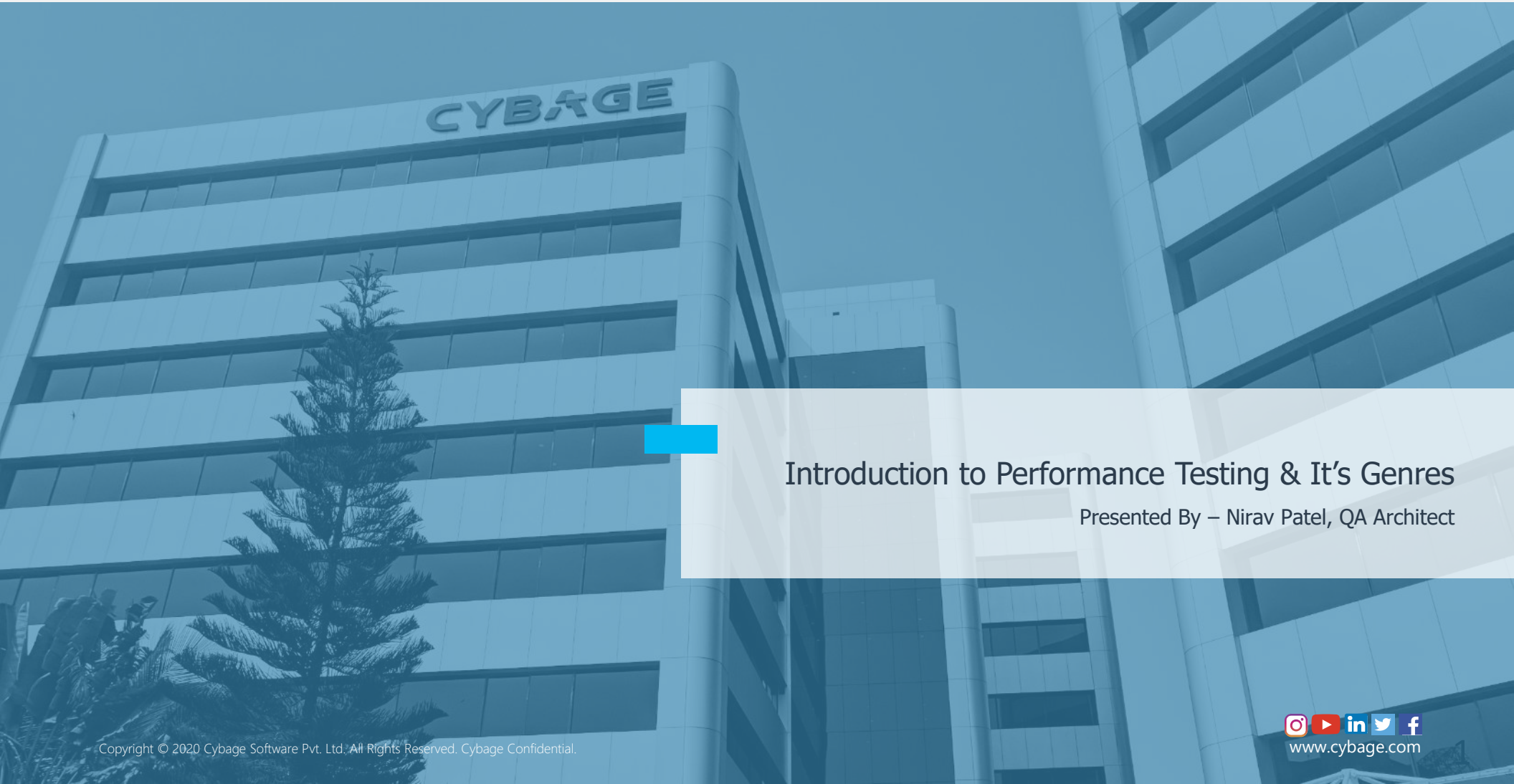


Introduction to Performance Testing & It's Genres

Presented By – Nirav Patel, QA Architect

Agenda

- What is Performance?
- User interaction & Common Website
- Why Performance Testing is Important?
- Performance Testing | Portfolio
- Terminologies
- Factors Affecting Applications Performance
- When To Execute?
- Attitude: Functional v/s Performance
- Performance Testing Approach

What is Performance?

How does one response when subjected to certain conditions.

Common Website

request

Internet

response

Accepting HTTP requests from web clients and serving the response,
Optional authorization request
Content compression to reduce size of response
Bandwidth throttling to limit the speed of responses
E.g. IIS, Apache, IBM HTTP

Makes applications available to multiple concurrent users.
Hosts an API to expose Business Logic and Business Processes
E.g. Web Sphere, JBoss

Provides database services like data storage ,concurrent access ,integrity of data and user authorization
E.g. SQL, ORACLE, MY-SQL

Why Performance Testing is Important?

- Engage customers with better website speed
- Time is money - The faster the website, the more revenue it will generate
- Resolve glitches before it goes to the market
- and many more...

Performance Testing | Portfolio

- Capacity
- Scalability
- Endurance
- Load
- Stress
- Spike

Capacity

- How much one can handle
- E.g. what is the capacity of the weight lifter i.e. how much weight he can lift easily?

20

200

'?'

5000

100

500

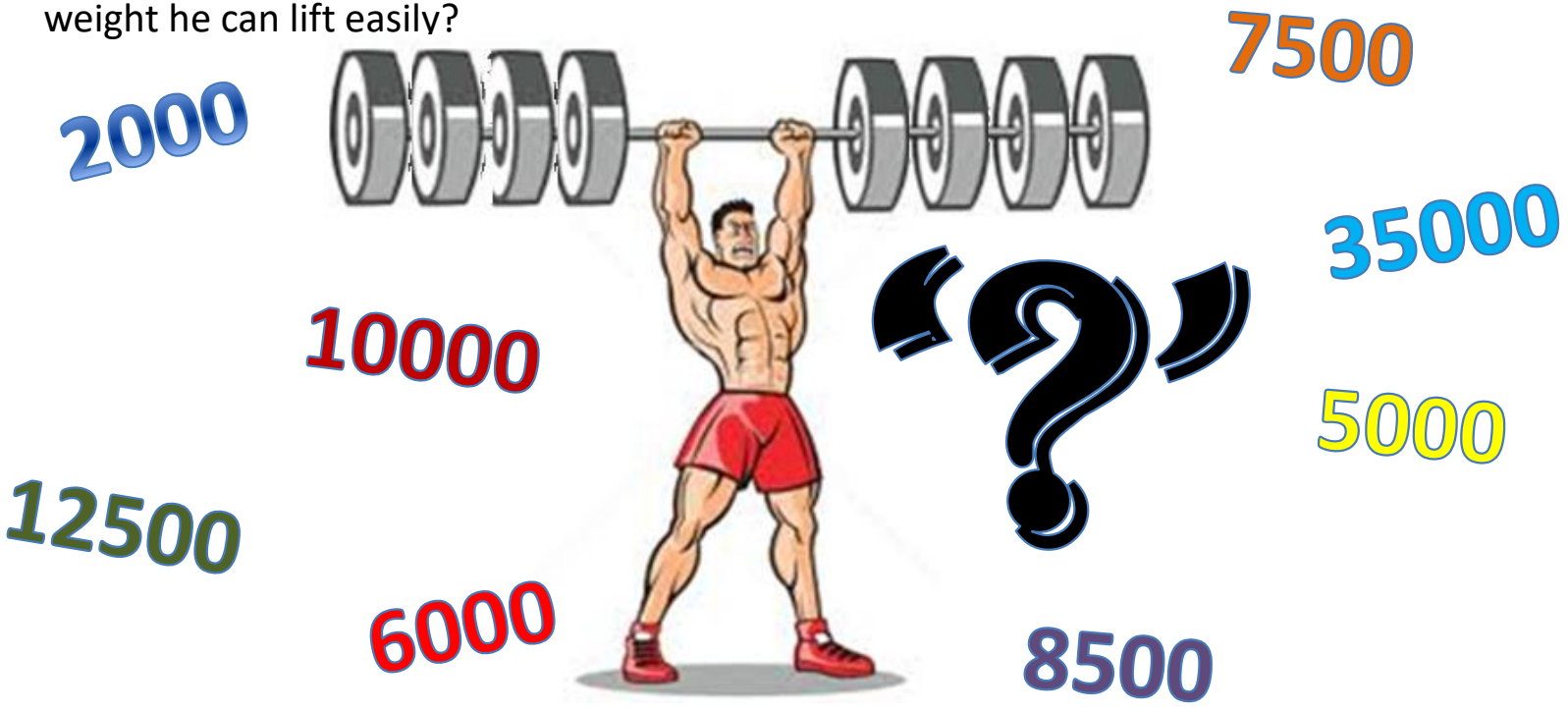
2500

10000

1000

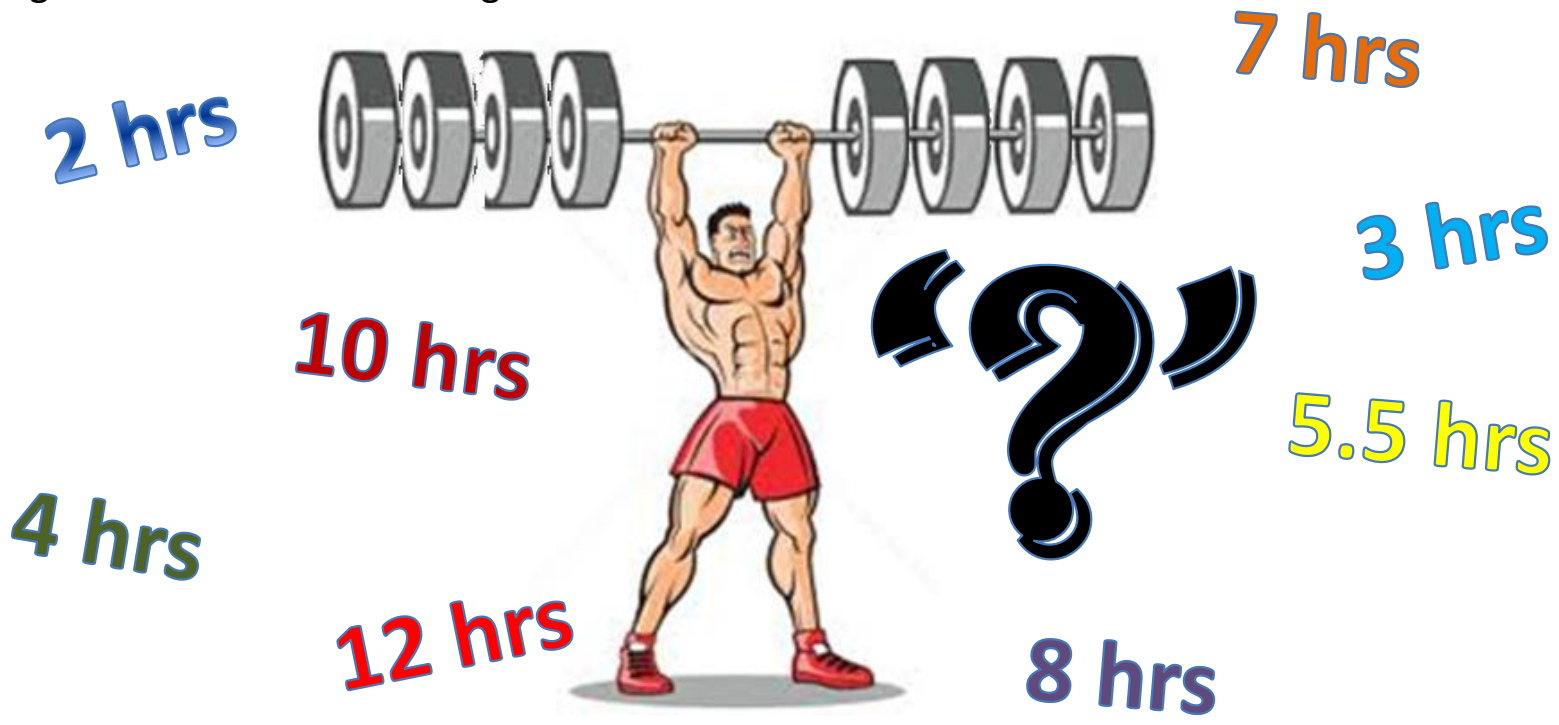
Scalability

- How much one can scale to or handle extra / additional
- E.g. what is the maximum capacity of the weight lifter i.e. how much max. weight he can lift easily?



Endurance

- How much one can handle continuously for a given timeframe
- E.g. can he lift the same weight for 8 hours in a stretch?



Load

- How does one behave when lifting load
- E.g. when lifting weights how does he feel or expresses

Stress

- How does one behave when subjected to extreme conditions?
- E.g. let’s give weight more than max he can lift and observe his behavior?

Spike

How does one behave with sudden increase in certain conditions?

Terminologies

- Virtual Users
- Think Time
- Throughput
- Transaction Mix
- User Mix
- Workload Mix
- Ramp Up and Ramp Down
- Peak Load/ Steady State
- Parameterization
- Correlation

Virtual Users

VU’s - Emulate the actions of human users by performing typical business actions

Why?

- Manual testing is difficult

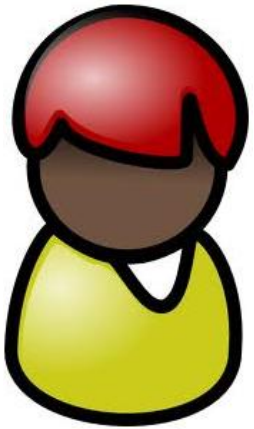
- When played back / Executed, the script interacts with the application just like a real user simultaneously

Think Time

"The time taken by the end user in between 2 transactions"

"Used to simulate human behavior that causes people to wait between interactions"

"Used to create more accurate load simulations"



- Open Hotel Booking Website
- Read the content **think time**
- Search for Rooms
- Go through the details of each rooms **think time**
- Select the Room
- Go through the offers/extras available **think time**
- Select the Offers
- Type the personal details **think time**
- Submit the details
- Verify the Reservation **think time**

Throughput

“The amount of transactions produced over time during test”

“No. of requests application can handle”

Requests / sec

Transactions / sec

Hits / sec

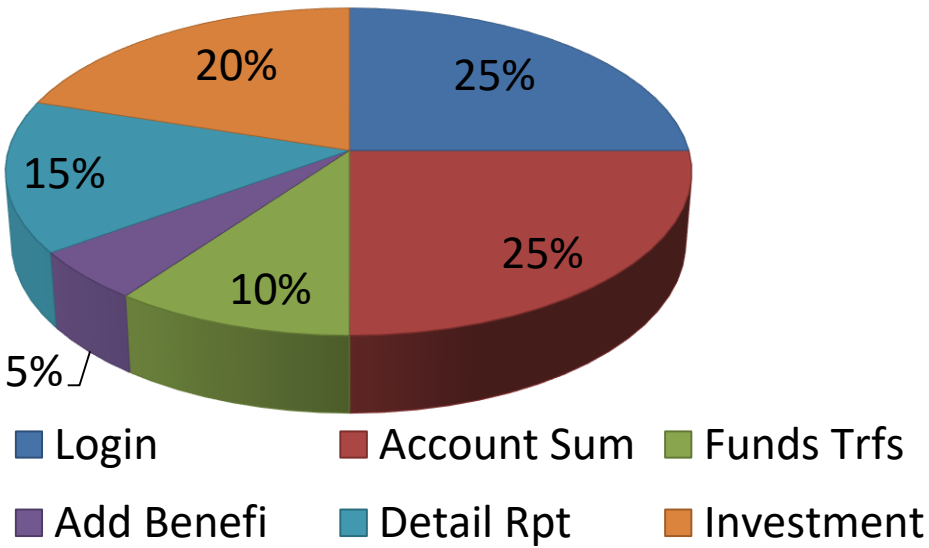
Factors which may affect Throughput are:

- *Response Time*
- *Server Configurations*
- *Hardware Consumption*
- *Unnecessary load on the server*
- *Background processes*
- *Application Architecture*
- *Network bandwidth etc.*

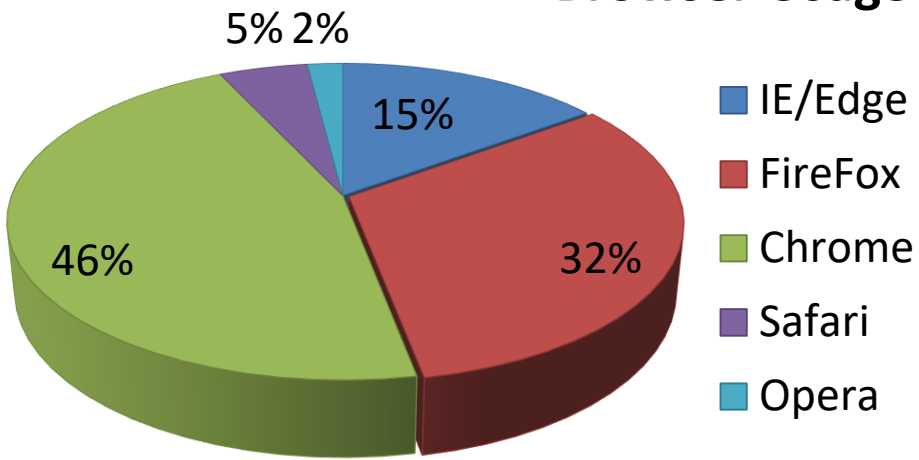
Transaction & Browser Mix

"Not all users do the same transactions, hence a mix of transactions & browsers needs to decided"

Banking Transactions

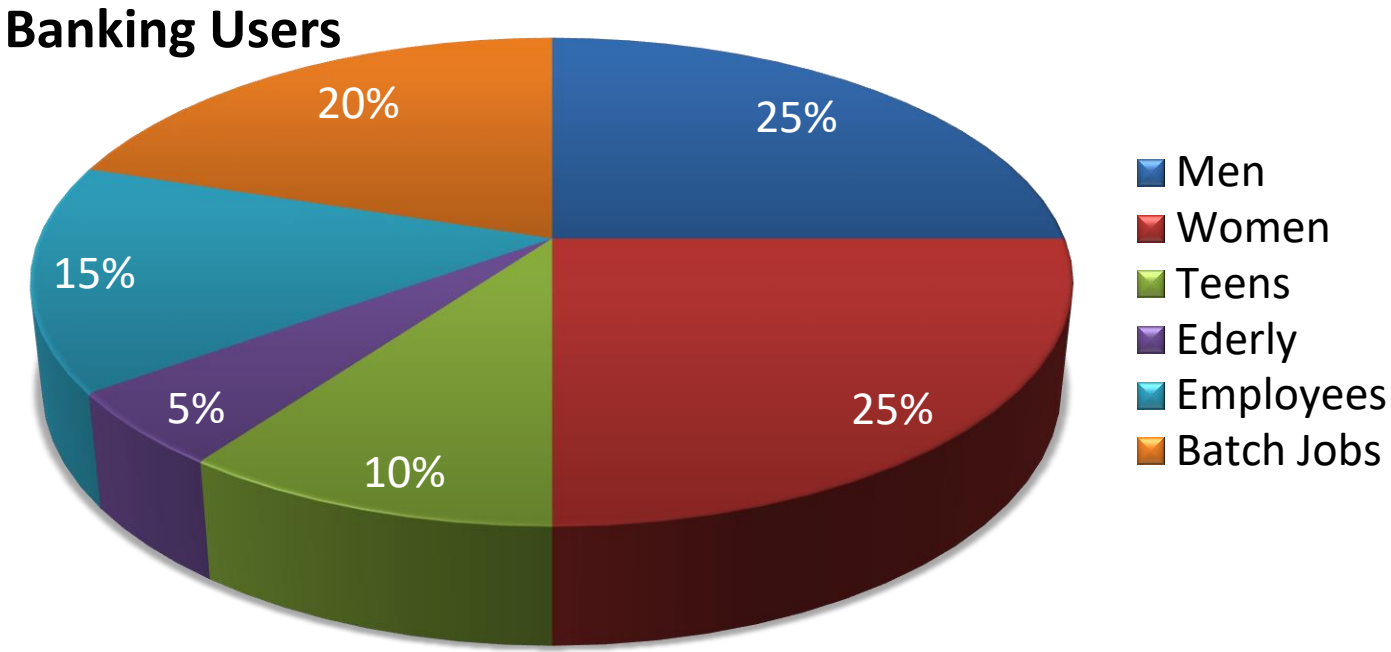


Browser Usage



User Mix

"Not all users are same i.e. speed of accessing application, performing transactions, nature of transactions, frequency of use etc."



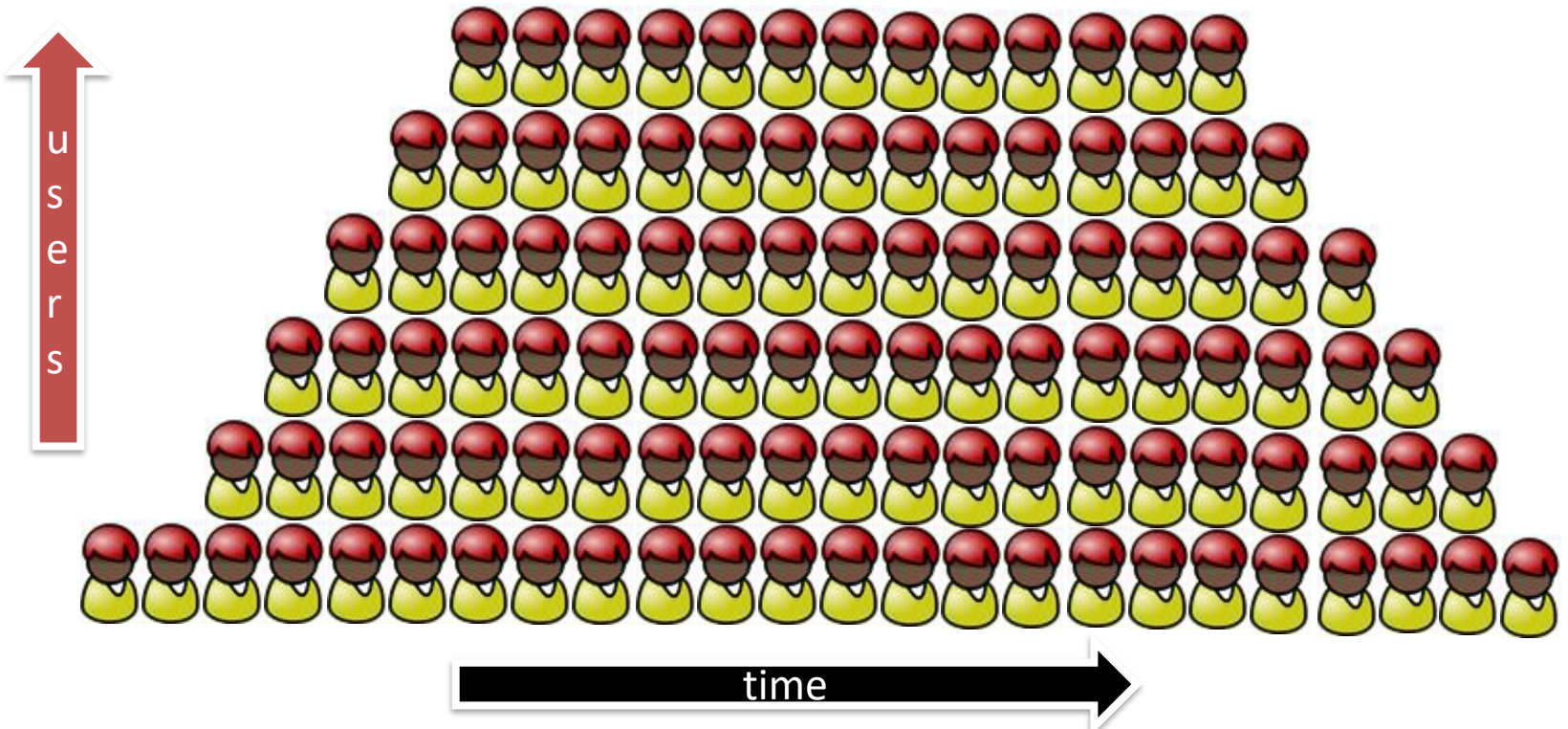
Workload Mix

"Includes the total number of users, concurrent active users, data volumes, transaction volumes and transaction mix"

Ramp up, Ramp Down & Steady State (Peak Load)

"Ramp Up is the process of initiating enough user sessions to drive the peak load"

"Ramp Down is the process of gracefully ending all the active user sessions"



High level script creation activity

- *Record User Action / Traffic*
- *Playback / Replay Recorded Traffic*
- *Enhance the script*
- *Use it for load generation*

Parameterization

What is Parameterization?

"Replacing a hard coded value with a variable / parameter and assigning data value from a data source say CSV / DB / XML"

Why Parameterization?

- *Use of Realistic Values rather than repeatable/ duplicate ones*
- *Avoid data caching during execution*
- *Ensure the requests are being processed by the Web / APP / DB servers*
- *Execute test with variety of data sets*

Correlation

What is* Correlation*?

*"Carrying forward **dynamic value(s)** received in previous Response to the Current / Next Request(s)."*

E.g. of dynamic values

- *Session Id,*
- *date time,*
- *record id,*
- *selected room id,*
- *authorization id,*
- *no. of records in a grid*

Why* Correlation*?

- *Avoid errors in script playback due to value mismatch*

Correlation

What is* Correlation*?

*"Carrying forward **dynamic value(s)** received in previous Response to the Current / Next Request(s)."*

E.g. of dynamic values

- *Session Id,*
- *date time,*
- *record id,*
- *selected room id,*
- *authorization id,*
- *no. of records in a grid*

Why* Correlation*?

- *Avoid errors in script playback due to value mismatch*

Why Performance Testing?

- Speed, response times
- Scalability, future growth
- Stability Issues
- Identify risks related to Revenue Loss
- Comparing different system configurations, which suites best
- Application exhibits desired performance within Budget
- Application exhibits desired performance before and after changes to the software
- Bottlenecks in application

Factors Affecting Applications Performance

There are several factors that affect performance.

- *Infrastructure i.e. server hardware configuration*
- *Network Topology, Bandwidth*
- *Software configuration*
- *Application Architecture*
- *Database Architecture / Design*
- *Third Party Integrations*
- *Scheduled backups, updates etc.*
- *many more . . .*

When To Execute?

"As Early as Possible in SDLC"

"Changes in Application / Hardware / Network / Configuration"

"New application hosted on the same server"

Pre-requisites:

- *Environment & Infrastructure*
- *Configuration*
- *Software Version*
- *Volume of Data*
- *Functional Stability*

Attitude: Functional v/s Performance

Are you able to achieve Funds Transfer?	How long does it take to achieve Funds Transfer?
Is the content displayed correctly as entered?	How long does it take to load the content? And is the time same for all the pages or vary as per page size?
Are there any JS errors when you click?	Do you get any server errors when you click?
Test the entire application functionality thoroughly	Test the most frequently used pages for performance
Log bugs for the functional defects and the developer will fix it	Log bugs for the performance issues and the team will analyze and tune
If the functionality is working in dev. environment, then the same will work in UAT	Performance results will vary from environment, you cannot use for relative comparison

Performance Testing Approach/Lifecycle

Discover	Analyze	Plan	Design and Build	Execution
• Application • Infrastructure • KPIs • SLA	• Configuration • Implementation • Data • Tool Feasibility	• Strategy • Scenarios • Monitoring	• Environment • Script and Data	• Execution • Analysis

Reporting

Performance Testing Tools

- Load Runner
- Apache JMeter
- Neoload
- many more…