Sample Title

Load testing a Magento store:

from selling the idea to delivering final report

Alec Bedzir

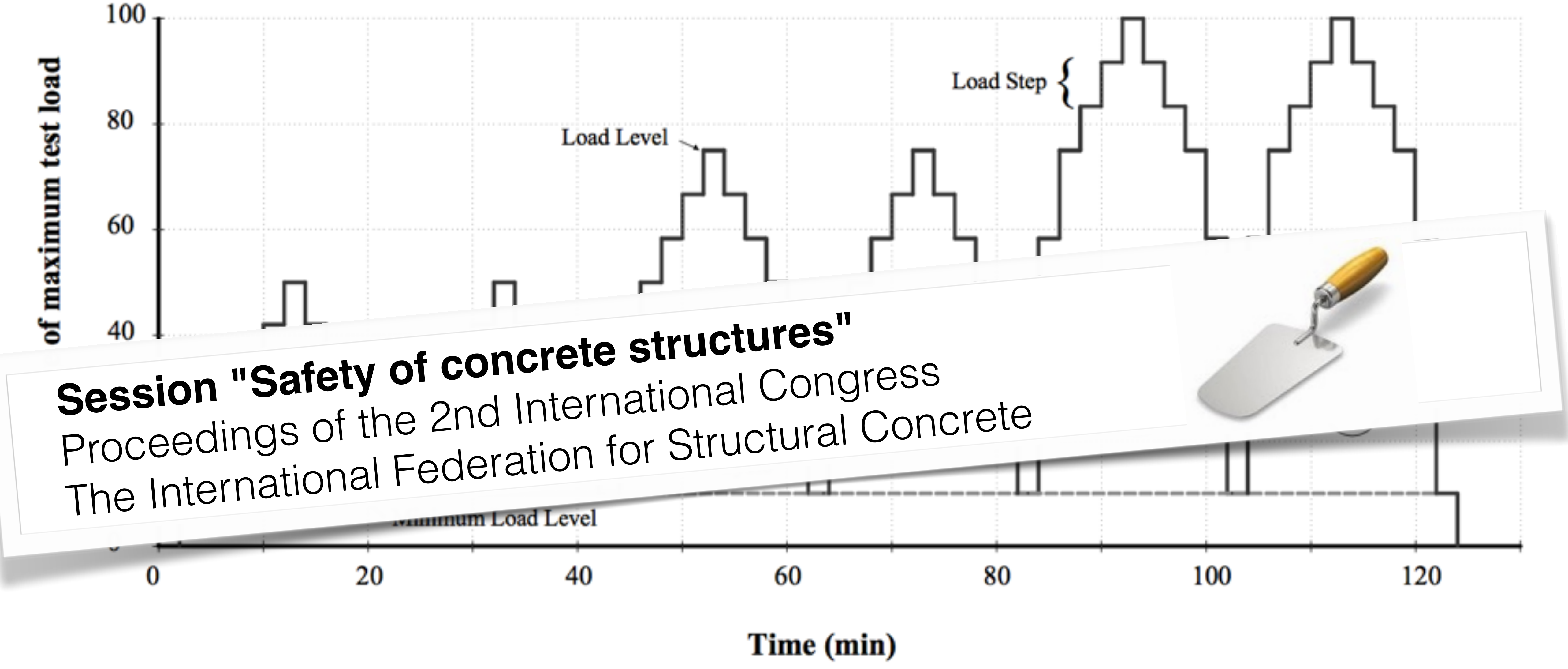
Head of Systems Architects - Vaimo

Outline

- Load Testing is an Old Story
- What is Load Testing
- Selling Points
- What it's not
- In-house Load Testing
    - Preparing the Load Test
    - Running the Load Test
    - Analysing and Creating a Report
- Vaimo Load Tests Statistics
- Load Test Monitoring with GA
- Distributed Load Testing
- Load Testing Market Trends

Load Testing is an Old Story

Load Testing is an Old Story



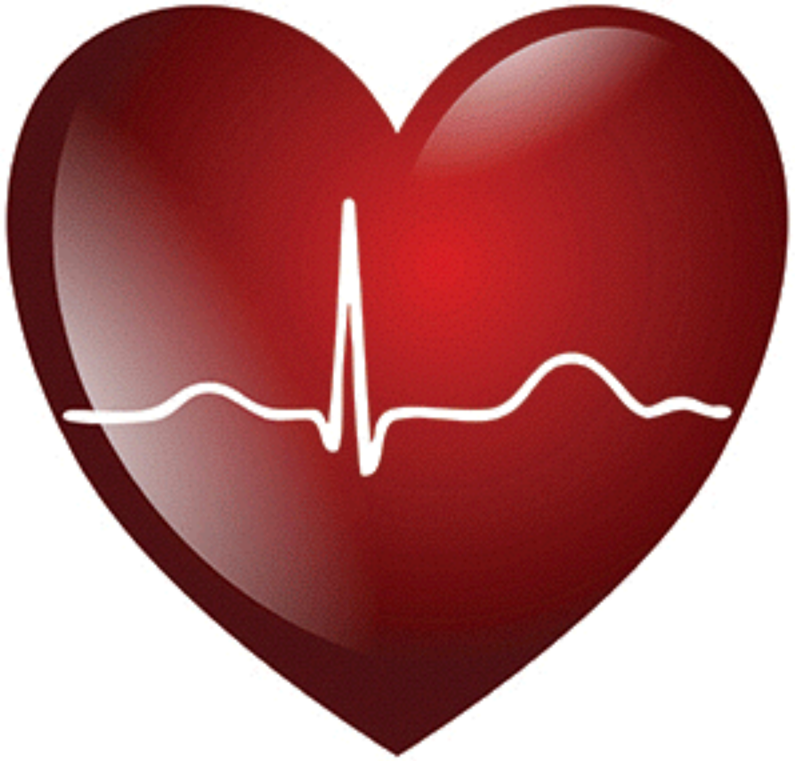
Session "Safety of concrete structures"
Proceedings of the 2nd International Congress
The International Federation for Structural Concrete

Fig. 1. Cyclic load test time history

Load Testing is an Old Story

Full-scale in-situ load tests on buildings in the US **dating back to 1910**

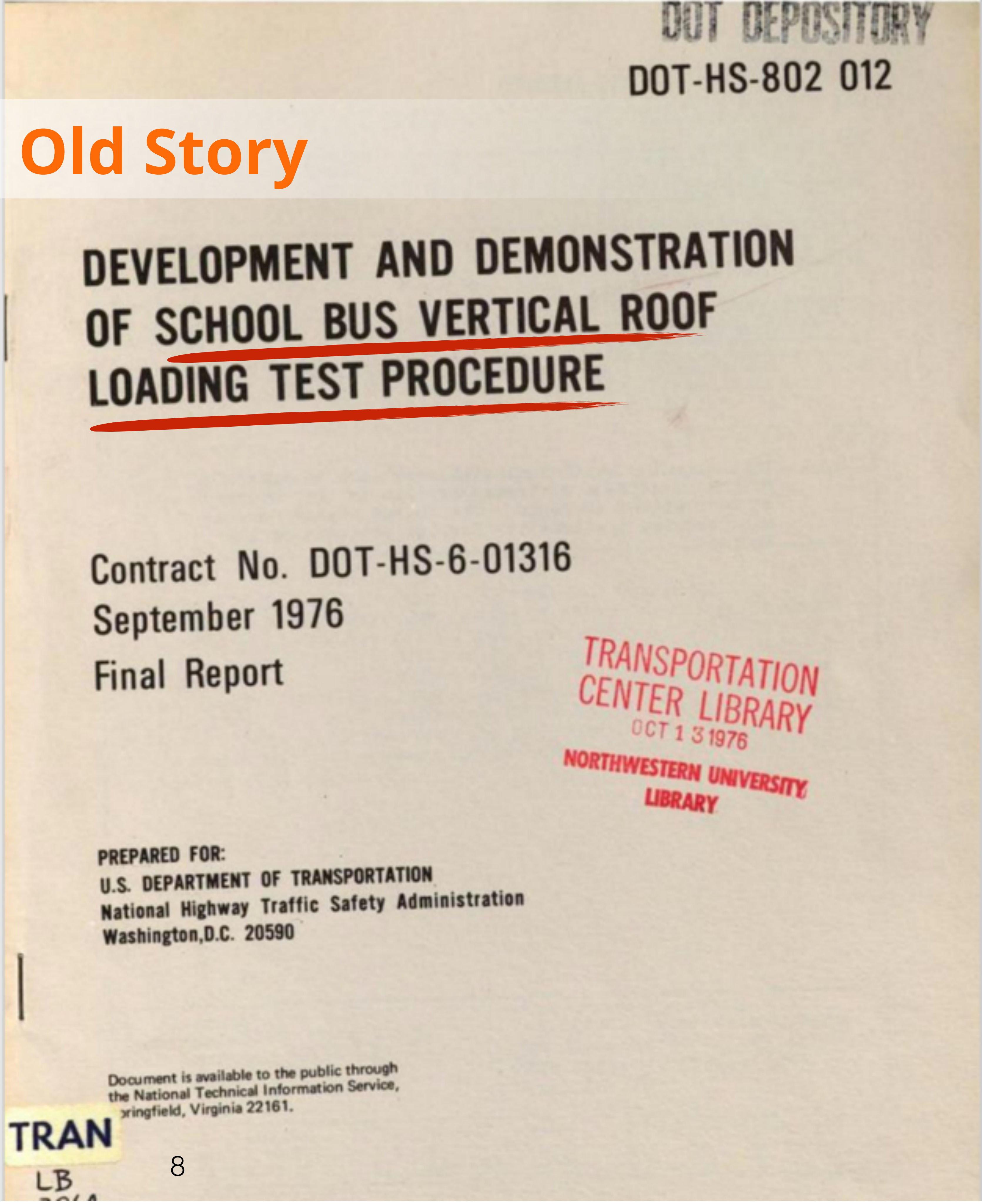
Load Testing is an Old Story



W

A **cardiac stress test** is a cardiological test that measures a heart's ability to respond to external stress

Load Testing is an Old Story



DOT-HS-802 012

DEVELOPMENT AND DEMONSTRATION OF SCHOOL BUS VERTICAL ROOF LOADING TEST PROCEDURE

Contract No. DOT-HS-6-01316
September 1976
Final Report

PREPARED FOR:
U.S. DEPARTMENT OF TRANSPORTATION
National Highway Traffic Safety Administration
Washington, D.C. 20590

Document is available to the public through the National Technical Information Service, ›pringfield, Virginia 22161.

What is Load Testing

What is Load Testing

msdn.microsoft.com

Load testing is a type of performance testing focused on identifying the maximum operating capacity of an application as well as any bottlenecks that might interfere with its operating at capacity.

What is Load Testing

W

Load testing is often **a synonym** for:

- concurrency testing
- software performance testing
- reliability testing
- volume testing

☆☆☆

What is Load Testing

Load Testing is often mixed up with **Stress Testing**

msdn.microsoft.com

Stress Testing is a type of performance testing focused on determining an application's robustness, availability, and reliability under extreme conditions.

Selling Points

Selling points

High level idea

achieving performance related Quality Attributes

Selling points

The ones from real life:

- verifying that web store meets KPI's
- determining real system capacities
- revealing operational issues and bottlenecks
- comparing systems performance-wise

☆☆☆

Selling points

Particular use cases (1):

confirm system can handle certain load thus proving it meets the requirements defined in KPI`s (or Performance Acceptance Criteria)

define the utmost load system can handle at the moment

determine the degraded performance modes (and indicate the load leading to each of these modes)

○ ○ ○

Selling points

Particular use cases (2):

snapshot development efforts and start tracking further development performance-wise

get insights before starting preparing the app for campaigns

confirm or refute reasoning for hardware chosen; benchmark various setups

compare systems performance-wise

○ ○ ○

Selling points

Particular use cases (3):

gather investigation material on slow application components and transactions

implement a self-check on project performance – steady graphs show all was architectured and implemented correctly

apply dynamic performance analysis for the system before taking over a project from another vendor

☆☆☆

What it's not

What it's not

Questions Load Testing doesn't answer:

- front-end and back-end code quality
- UI performance
- presence of bugs
- compatibility (with standards, browsers etc.)
- usability
- security
- privacy

☆☆☆

In-house Load Testing

In-house Load Testing

Preparing the load test (1):

1. Defining KPIs

   (Performance Acceptance Criteria)
   These can be response time, throughput, maximum concurrent users etc.

2. Designing scenarios

   User journeys specific for the application, balanced workload

In-house Load Testing

Preparing the load test (2):

3 Planning the load

Define the number or load test rounds and the load level for each round

4 Choosing the metrics

Network, system, stack, application and business metrics

☆☆☆

In-house Load Testing

Running the load test (1):

1. Preparing the environment

   Clone production environment or run the test on production directly

2. Generating the load

   Run the test plan

In-house Load Testing

Running the load test (2):

3

Cleaning up

Restore original database reverting all the test-specific changes to the environment

☆☆☆

In-house Load Testing

Analysing and creating a report (1):

1 Collecting metrics

Add all the metrics to the report (charts, tables, screenshots etc.)

2 Concluding

Provide analysis of metrics, intermediate and final summary

In-house Load Testing

Analysing and creating a report (2):

3

Creating a report

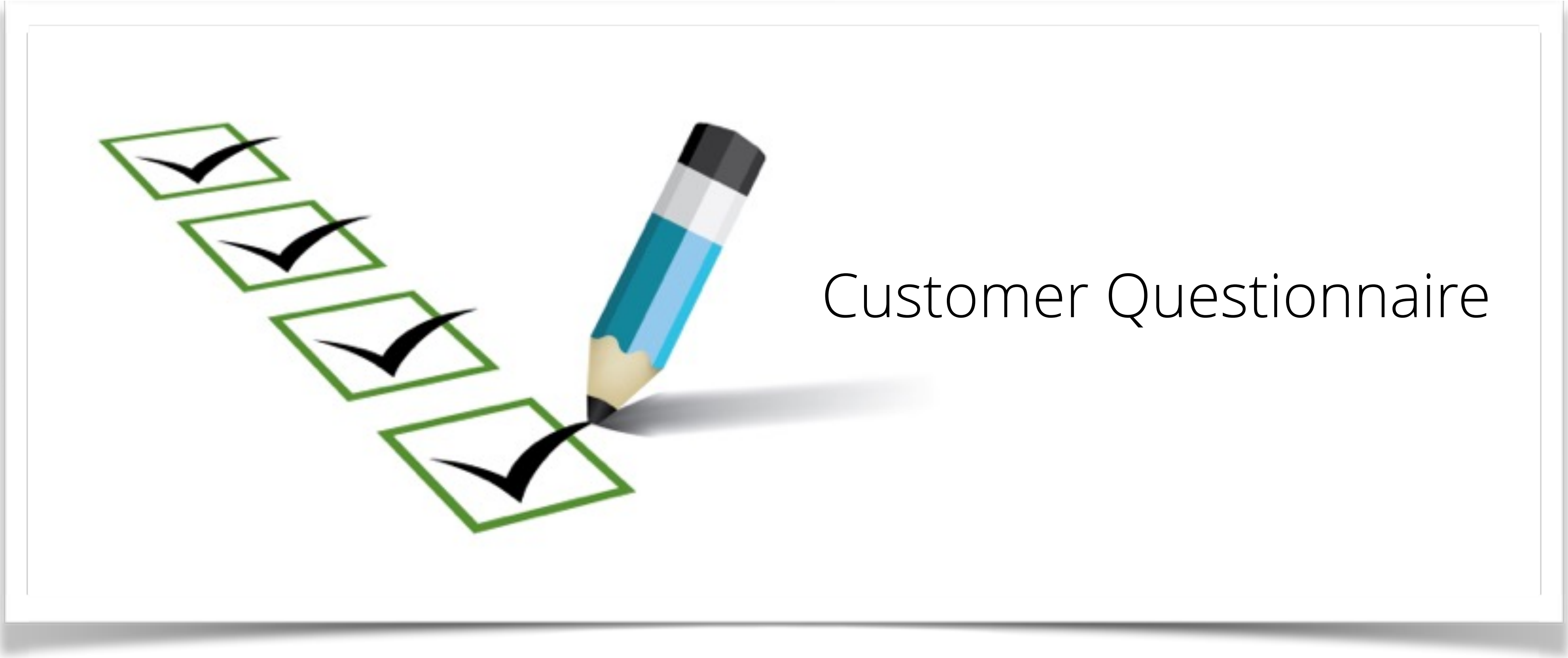
Deliver the final document containing the whole load test process description and results

☆☆☆

In-house Load Testing

Preparing the load test

KPI`s Scenarios Planning the load Choosing metrics



Customer Questionnaire

In-house Load Testing

Preparing the load test

Customer Questionnaire (1):

- Load Testing goals
- planned date or deadline
- environment to Load Test
    - production environment
    - identical to production (built for load test)
    - any existing non-production environment

In-house Load Testing

Preparing the load test

Customer Questionnaire (2):

- KPI's (performance criteria)
  - end-user response time
  - maximum concurrent users
  - throughput
  - resources utilization
  - business metrics

In-house Load Testing

Preparing the load test

Customer Questionnaire (3):

- sales funnel to be emulated

- 62% abandoned cart: adding items and abandoning the cart
- 30% browsing only (with and without layered navigation)
- 4% browsing, adding items to cart, checking out as a guest
- 4% browsing, adding, checking out as a registered customer

In-house Load Testing

Preparing the load test

Customer Questionnaire (4):

- scenarios (user journeys)
- activities

- Homepage view
- Category view / filtered / sorted / N-page in pagination / lazy loader
- Product view
- Search result non-empty view / empty
- Add to cart action
- Cart view
- Wishlist actions and views
- Comparison list actions and views
- Checkout actions and views
- Review action, newsletter action
- Login action / logout action
- Customer account related page views
- Static pages
- Sitemap containing endpoints
- Custom transactions (project specific)

In-house Load Testing

Simple scenario example: checkout with one product

- Open Homepage
- Open Random Category Page
- Open Random Product and add it to Shopping Cart
- Open Shopping Cart Page
- Perform checkout as guest / registered customer
- Order Success Page

In-house Load Testing

thinking of embedded resources

Ensure load testing tool retrieves all the embedded resources from the page.

JMeter > HTTP Request Defaults:

- Retrieve All Embedded Resources (enable)
- Use concurrent pool (enable)

In-house Load Testing

thinking of AJAX requests

Ensure Javascript triggered requests are emulated

Browser-like load testing tool:

- available out of box

JMeter based load testing tool:

- requests in parallel threads
- using the JMeter WebDriver Sampler
- scripting for JSR223 Sampler
- developing a custom Sampler

In-house Load Testing

insights into most popular, time consuming transactions

Google Analytics

- Behaviour > Site Content > All Pages
- Behaviour > Site Speed > Page Timings

New Relic. APM

- Monitoring > Transactions

New Relic. INSIGHTS™

- NRQL: SELECT uniqueCount(session), average(backendDuration) FROM PageView where appName='...' since X hours ago FACET name

In-house Load Testing

recommendations on designing scenarios (1):

- think of future tests beforehand – stick to generic scenarios
- utilize numerous entities in your test data sets (customers to place orders, categories to visit etc.)
- emulate wrong customer behaviour (mistyping passwords, emails etc.)
- configure different think time for your customer cohorts (new / returning customers, quick / unhurried buyers)

○ ○ ○

In-house Load Testing

recommendations on designing scenarios (2):

- consider launching background processes emulating Magento back-end and scheduled activities
- avoid load testing 3rd party services (CDN, payment etc.) unless there's a certain reason for doing that

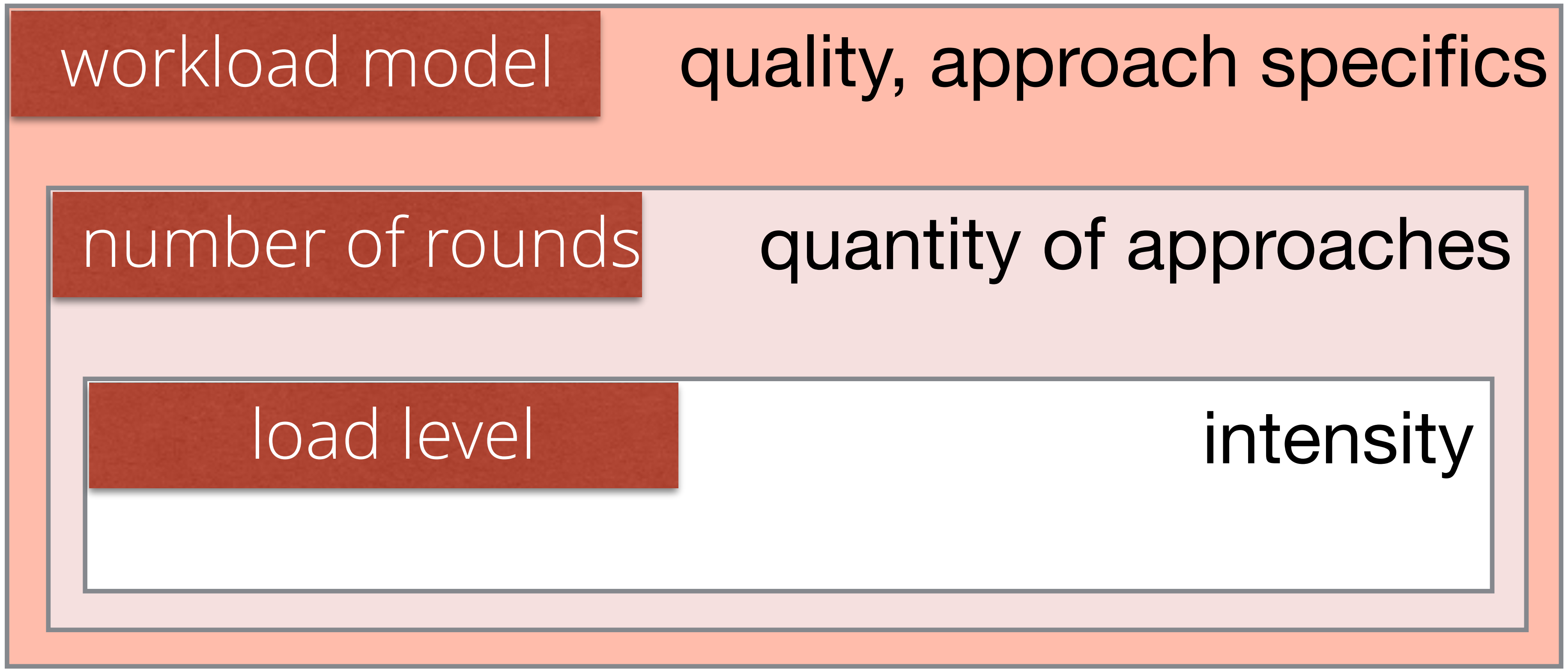
☆☆☆

In-house Load Testing

Preparing the load test

Customer Questionnaire (5):

- planing the load



workload model

quality, approach specifics

number of rounds

quantity of approaches

load level

intensity

In-house Load Testing

Preparing the load test

Customer Questionnaire (5):

workload model

- delays (think time)
- session average duration
- overlapping sessions
- customer cohorts
- increase / decrease in traffic
- operations on the environment

In-house Load Testing

Preparing the load test

Customer Questionnaire (5):

load level

- concurrent users per unit of time
- distribution of work

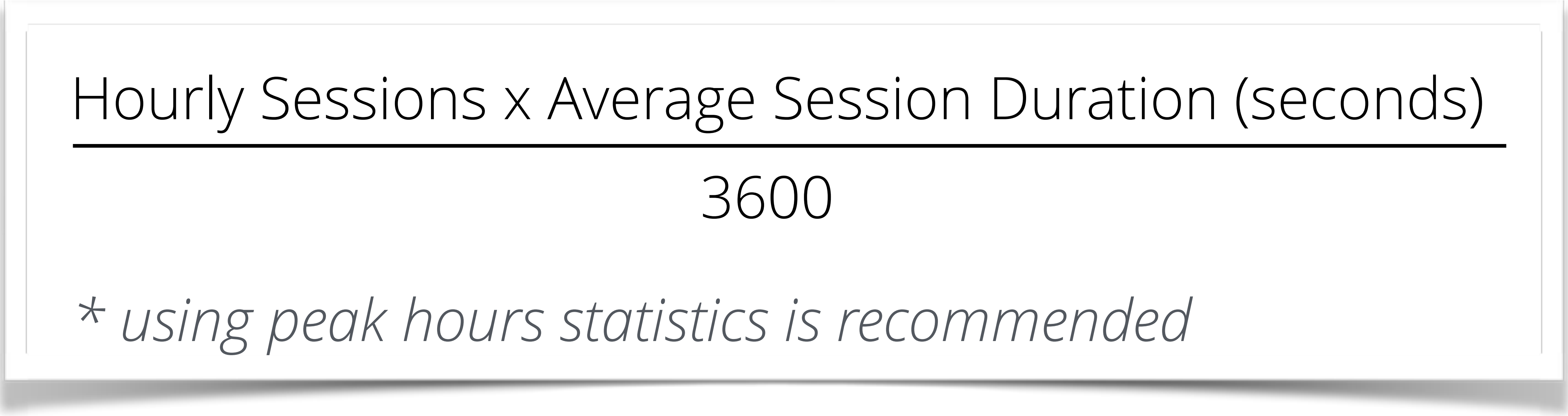
☆☆☆

In-house Load Testing

planning the load properly (1):

Concurrency

Google Analytics Audience > Overview (Sessions)



$$\frac{\text{Hourly Sessions x Average Session Duration (seconds)}}{3600}$$

** using peak hours statistics is recommended*

In-house Load Testing

planning the load properly (2):

Preliminary Adjusting Load Test Round

1. initiate a short preliminary round to test the planned load
2. use your load level figures (concurrency, think time etc.) to emulate known environment load
3. compare the charts of monitoring and analytics to those representing the known load
4. adjust the load level figures if necessary

☆☆☆

In-house Load Testing

planning the load properly (3):

Do not change the workload model across load testing rounds.

While the scale is supposed to be changed the workload model (incl. scenarios, requests distribution, operations on the environment etc.) should remain intact

In-house Load Testing

Preparing the load test

Customer Questionnaire (7):

- metrics levels

- Network
- System
- Platform
- Application
- Service
- Business

In-house Load Testing

Preparing the load test

Customer Questionnaire (7):

- metrics

external

- response time (elapsed, latency, connection)
- throughput
- orders placed

internal

- apdex score
- throughput
- slowest transactions
- transactions with highest throughput
- FPC hit rate
- Load average, Memory
- slow DB queries
- SQL volume (reads, writes)
- web server throughput
- command rate for cache engine
- errors

In-house Load Testing

Running the load test

Environment

Load

Cleaning up

In-house Load Testing

Running the load test (example of running on Production) (1):

- clone existing Production DB (sanitize if required)
- ensure monitoring and analytics are in place
- pause your ping, availability or health-check services
- enable Maintenance mode (forward load test tool IP`s)
- ensure no integration feeds are running
- switch to DB clone and make environment operating
- pause all the integrations

○ ○ ○

In-house Load Testing

Running the load test (example of running on Production) (2):

- ensure any system-customer communication is disabled
- adjust the flows if needed
  (enable Cash on Delivery, bypass single sign-on etc.)
- backup critical log files
- issue a "load test start" marker in a web server and application logs
- load test the environment using your favourite tool
- backup gathered web server and application logs

In-house Load Testing

Running the load test (example of running on Production) (3):

- clean up and restore critical logs from the backup
- switch application back to the original DB
- ensure environment operates normally
- enable system-customer communication
- enable all the integrations and remove the MNT flag
- backup monitoring and analytics load test data ASAP
- remove load test DB after it has been analysed

☆☆☆

In-house Load Testing

consider the following important points:

- Magento cron should keep running throughout the load test
  (should not be used to pause integrations and system-customer communication)
- resumed integrations can result in performance spike – wait before you let customers in
- monitoring and analytics will alarm a lot which could distort your monthly SLA report

In-house Load Testing

recommendations (1):

- flows adjustment should not affect chosen metrics
- consider increase and decrease in traffic as very important parts of your Workload: start small and change gradually
- load test round duration:
  warm up time + *time for analysing the system under load*
- *time for analysing the system under load*: is a time for either proving stability or identifying degraded modes clearly

○ ○ ○

In-house Load Testing

recommendations (2):

- load test rounds should last for the same amount of time
- each load test round should start with exactly the same state of the system
- make pauses between load test rounds for monitoring and analytics to calm down
- use *distributed load testing* in case significant load level is required

☆☆☆

In-house Load Testing

Analysing and creating a report

Collecting metrics

Concluding

Creating a report

In-house Load Testing

Analysing and creating a report

Collecting metrics:

external

- response time (elapsed, latency, connection)
- throughput
- orders placed

internal

- apdex score
- throughput
- slowest transactions
- transactions with highest throughput
- FPC hit rate
- Load average, Memory
- slow DB queries
- SQL volume (reads, writes)
- web server throughput
- command rate for cache engine
- errors

In-house Load Testing

Analysing and creating a report

Report sections (1):

- Terms
- Load Test Goals
- Environment
- KPIs
- Metrics
- Scenarios

○ ○ ○

In-house Load Testing

Analysing and creating a report

Report sections (2):

- Workload model
- Load Test Rounds: duration and load levels
- Summary
  - Analysis of external metrics
  - Analysis of internal metrics
  - Final conclusions

○ ○ ○

In-house Load Testing

Analysing and creating a report

Report sections (3):

- Metrics collected
  - Metrics from Round 1
  - Metrics from Round 2
  - Metrics from Round N
  - Metrics for application with no extra load

☆☆☆

In-house Load Testing

Hints on summary (1):

- compare retrieved figures to the ones from Magento benchmarks (either done by 3rd party vendors or internal)
- outline the degraded performance modes if any and load levels leading to them
- describe system performance and load level when thresholds identified in objectives (if any) are crossed

○ ○ ○

In-house Load Testing

Hints on summary (2):

- make a note of load level and system performance when your company SLA limits are reached
- outline all the bottlenecks detected
- for metrics where applicable compose lists of top / worst performing transactions (operations, pages etc.)
- create a list of most important optimization actions and most critical areas to be optimized

○ ○ ○

In-house Load Testing

Hints on summary (3):

- describe performance fluctuation: explain as many chart changes as possible binding them to load test actions
- make a note of time it takes for system to warm up, stabilize, calm down etc.

☆☆☆

Vaimo Load Testing Statistics

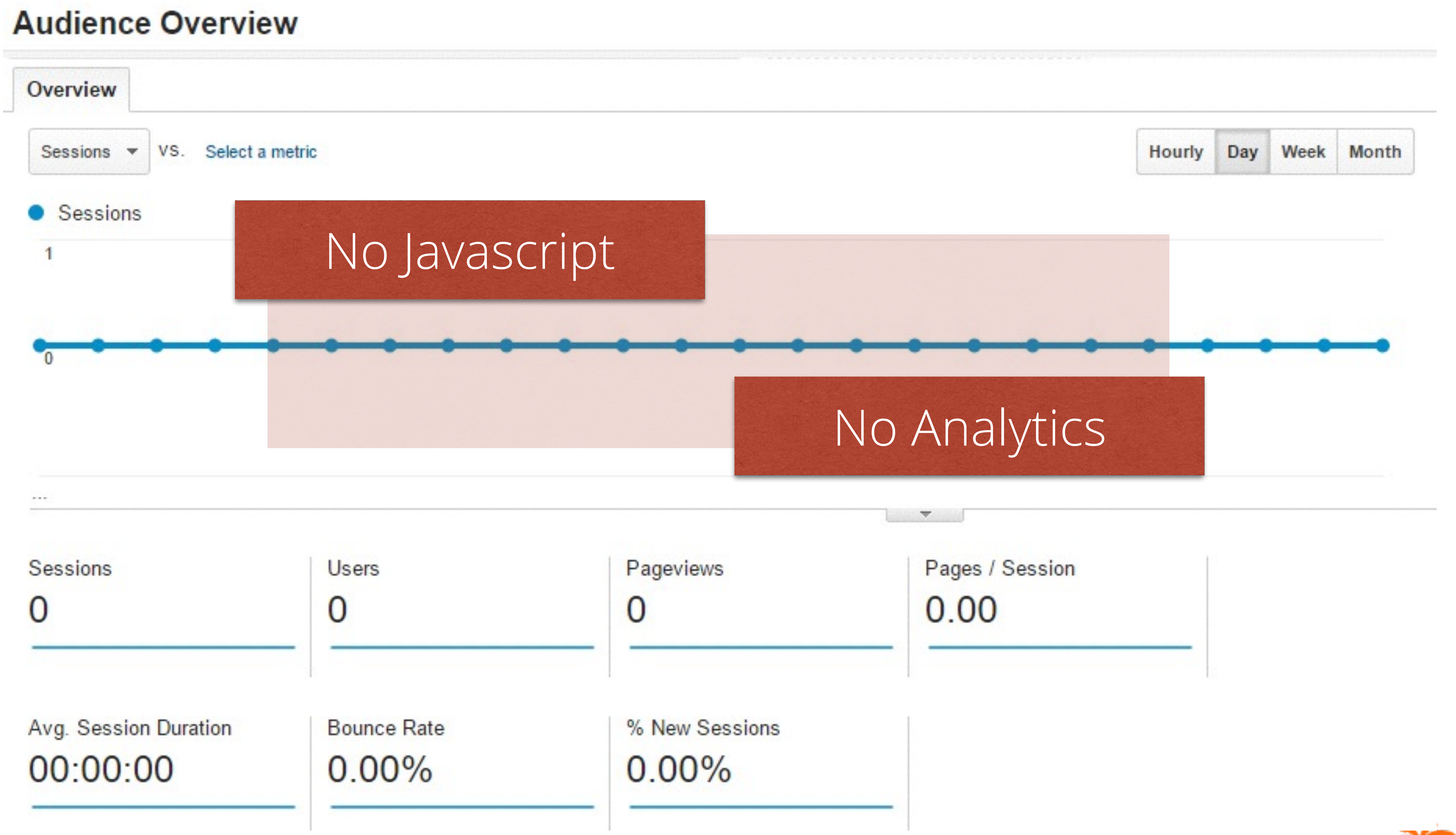
Vaimo load tests statistics

- load test round time is 20 min
- average load test is 1 preliminary and 2-3 real rounds
- overall load test duration is about 2-2.5 hours
- maintenance window is 3-4 hours
- time spent on a load test from idea to report is about 25-30 hours
- load testing tool: JMeter, monitoring: NewRelic, analytics: GA
- load test team: 3 members

☆☆☆

Load Test Monitoring with GA

Load Test Monitoring with GA



No Javascript

No Analytics

Audience Overview

Overview

Sessions
0

Users
0

Pageviews
0

Pages / Session
0.00

Avg. Session Duration
00:00:00

Bounce Rate
0.00%

% New Sessions
0.00%

Load Test Monitoring with GA

GET to https://www.google-analytics.com/collect with params:

Param	Value	Description
cid	123456789	Unique Session ID
dl	http://site.com/product-a	Page URL
dt	Product A	Page Title
t	pageview	Transaction Type
tid	UA-12345678-9	GA tracking code
gtm	GTM-ABCDEF	GTM Container ID
v	1	GA API Version

Load Test Monitoring with GA

JMeter Implementation

- Adding BeanShell Preprocessor to each Thread Group
- Generating *cid* (Unique Session ID) using:

```
...
int threadNo = ctx.getThreadNum()+1;
int threadGroupBase = 123456789;
int uniqueId = threadNo + threadGroupBase;
vars.put("uniqueId", Integer.toString(uniqueId));
...
```

Load Test Monitoring with GA

JMeter Implementation

- Using Regular Expression Extractor for *dt* (Page Title)
- Using known URLs of target pages for *dl* (Page URL)
- Using hardcoded values for
  - *t* (Transaction Type)
  - *tid* (GA tracking code)
  - *gtm* (GTM Container ID
  - *v* (GA API Version)
- Adding an extra HTTP Request (performing GET to GA endpoint) for each existing page request

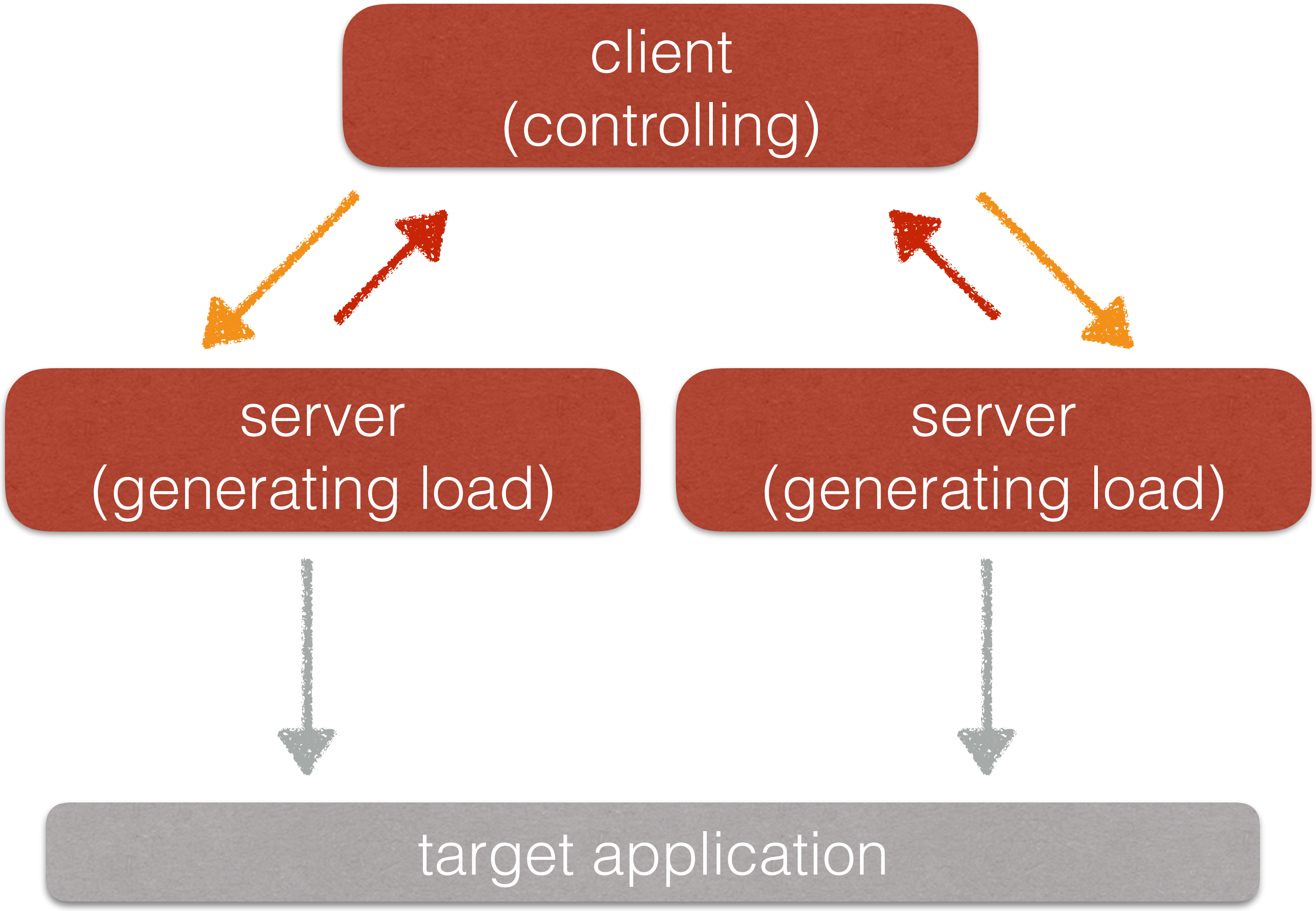
Distributed Load Testing

Distributed load testing with JMeter

—
starting the test

—
test results

—
test plan requests



Distributed load testing with JMeter

- install the same versions of JMeter for both client and servers
- configure and launch servers on different machines:

```
# for each server:

bin/jmeter-server \
  -Djava.rmi.server.hostname=127.0.0.1 \
  -Jserver_port=[port1] \
  -Jserver.rmi.localport=[port2]
```

Distributed load testing with JMeter

```
# example:

# Server 1
bin/jmeter-server -Djava.rmi.server.hostname=127.0.0.1 \
    -Jserver_port=10101 -Jserver.rmi.localport=10102

# Server 2
bin/jmeter-server -Djava.rmi.server.hostname=127.0.0.1 \
    -Jserver_port=20101 -Jserver.rmi.localport=20102
```

Distributed load testing with JMeter

- configure and launch a client:

```
bin/jmeter.sh \
  -Djava.rmi.server.hostname=127.0.0.1 \
  -Jremote_hosts="\
  127.0.0.1:[server_port on server #1],\
  127.0.0.1:[server_port on server #2]" \
  -Jclient.rmi.localport=[port]
```

* specify all the jmeter servers/ports separated by comma

* parameter "client.rmi.localport" is a randomly chosen port

Distributed load testing with JMeter

```
# example:

bin/jmeter.sh \
  -Djava.rmi.server.hostname=127.0.0.1 \
  -Jremote_hosts="127.0.0.1:10101,127.0.0.1:20101" \
  -Jclient.rmi.localport=10100
```

Distributed load testing with JMeter

- start SSH tunnels

```
# for each server:

ssh -vN \
    -L [server_port]:127.0.0.1:[server_port] \
    -L [server.rmi.localport]:127.0.0.1:[server.rmi.localport] \
    -R [client.rmi.localport]:127.0.0.1:[client.rmi.localport] \
    user@server
```

Distributed load testing with JMeter

```
#example

# Terminal 1
ssh -vN -L 10101:127.0.0.1:10101 \
  -L 10102:127.0.0.1:10102 -R 10100:127.0.0.1:10100 \
  user@server-A

# Terminal 2
ssh -vN -L 20101:127.0.0.1:20101 \
  -L 20102:127.0.0.1:20102 -R 10100:127.0.0.1:10100 \
  user@server-B
```

Distributed load testing with JMeter

- start testing:

Run -> Remote start all
to start testing from remote servers

Load Testing Market Trends

Load Testing Market Trends

Trends (1):

- virtual users based pricing models
- functional testing instead of load testing
- browser emulation
- scenario recorders
- custom scripting support

○ ○ ○

Load Testing Market Trends

Trends (2):

- shift to continuous load testing
- multi-geo testing
- mobile testing: simulating mobile network conditions and recording mobile user scenarios
- server side monitoring integration
- scheduling and automation of load tests

☆☆☆

Resources

Resources

- **Performance Testing Guidance for Web Applications**
  by Microsoft Corporation
  Publisher: Microsoft Press. November 2007
  https://www.safaribooksonline.com/library/view/performance-testing-guidance/9780735625709/

- **Microsoft Developer Network materials**
  https://msdn.microsoft.com

- **Calculating the number of concurrent users to test**
  Knowledge base at support.loadimpact.com
  http://support.loadimpact.com/knowledgebase/articles/265461-calculating-the-number-of-concurrent-users-to-test

Resources

- **How to Load Test AJAX/XHR Enabled Sites With JMeter**
  by Dmitri Tikhanski. July 29, 2014
  Blog of BlazeMeter.com
  https://www.blazemeter.com/blog/how-load-test-ajaxxhr-enabled-sites-jmeter

- **JMeter - 2.9 - (Remote Test|Distributed testing)**
  by Nicolas Gerard
  http://gerardnico.com/wiki/jmeter/remote

- **JMeter Distributed Testing Step-by-step**
  User manual at jmeter.apache.org
  http://jmeter.apache.org/usermanual/jmeter_distributed_testing_step_by_step.pdf

Resources

- **How to run jmeter over ssh tunnel**
  by Ionel Cristian Mărieș. February 16, 2012
  Blog at ionelmc.ro
  https://blog.ionelmc.ro/2012/02/16/how-to-run-jmeter-over-ssh-tunnel/

- **Magento Performance Toolkit**
  by William Harvey at MagentoLive Germany. 2014
  http://bit.ly/1S5tRVr

- **Track Non-JavaScript Visits In Google Analytics**
  by Simo Ahava. Febdruary 09, 2015
  http://www.simoahava.com/analytics/track-non-javascript-visits-google-analytics/

Resources

- **In-situ Load Test: a Case Study**
  Fédération Internationale du Béton.
  Proceedings of the 2nd International Congress
  June 5-8, 2006 – Naples, Italy
  http://rb2c.mst.edu/media/research/rb2c/documents/C9.pdf

- **Guidelines for rapid load testing of concrete structural members**
  by Matthew Mettemeyer and Antonio Nanni
  University of Missouri-Rolla. 2000
  http://www6.miami.edu/cici/Documents/Projects/1999_2000/P%201999-2000_R10.pdf

Questions?

Load testing a Magento store:

from selling the idea to delivering final report

Alec Bedzir

Head of Systems Architects at Vaimo

alec.bedzir@vaimo.com

Thank You!

DevelopersParadise

2016 / Opatija / Croatia