

183.290 Software Testing



Module: ???

Gastvortrag

WS 2020/21

Klemens Loschy (klemens.loschy@seqis.com)
Principal Consultant, Teamlead
SEQIS GmbH (<https://www.seqis.com>)

peso@inso.tuwien.ac.at
<https://peso.inso.tuwien.ac.at>



INSO - Industrial Software

Institut für Rechnergestützte Automation | Fakultät für Informatik | Technische Universität Wien

Before we actually start ...

Kahoot!

- 7 questions
- 60 seconds to answer
- Sweets & honor for the winner!
- Join @ <https://kahoot.it> Game Pin: 5980083
- **IMPORTANT:** the winner needs to take a picture of his screen!

Kahoot!



Module: Efficiency Testing

Gastvortrag

WS 2020/21

Klemens Loschy (klemens.loschy@seqis.com)
Principal Consultant, Teamlead
SEQIS GmbH (<https://www.seqis.com>)

peso@inso.tuwien.ac.at
<https://peso.inso.tuwien.ac.at>



INSO - Industrial Software

Institut für Rechnergestützte Automation | Fakultät für Informatik | Technische Universität Wien

Today's Topics

- **Why Efficiency Testing (ET)?**
- **Two types of Requirements**
- **Some challenges of Efficiency Testing**
- **Different (technical) Load Testing approaches**
- **Tools and Common Usage**
- **Hands on: Jira Load Testing with NeoLoad**

Efficiency Testing

- *Testing to determine the efficiency of a software product*

Performance Testing

- *... determine the performance load)*

Efficiency: The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions. (ISTQB, ISO 9126 is very similar)

Load Testing

- *... type of performance testing with increased load*

Stress (and recovery) Testing

- *... type of performance testing at or beyond the limits*

Spike Testing

- *... type of performance testing short bursts of high load*

Why Efficiency Testing?

OnlineBanking und BusinessNet

OnlineBanking und BusinessNet stehen Ihnen im Moment leider nicht zur Verfügung.
Derzeit werden dringende technische Anpassungen durchgeführt.
Wir ersuchen Sie um etwas Geduld und danken für Ihr Verständnis.

Peinlich!

16. August 2011 14:16

Spritpreis-Datenbank funktioniert nicht

OnlineBanking und BusinessNet

Mit Antenne Salzburg finden Sie trotzdem Spritpreise

We ap
This is
We ask

„Don't Smoke-Volksbegehren: Erneut IT-Probleme bei Unterstützungserklärungen“ (Die Presse)



Foto © APA



„Sommerzeit abschaffen? Umfrage-Server kollabiert“ (Die Krone)



OnlineBanking und BusinessNet

OnlineBanking und BusinessNet stehen Ihnen im Moment leider nicht zur Verfügung.

Derzeit werden dringende technische Anpassungen durchgeführt.

Wir ersuchen Sie um etwas Geduld und danken für Ihr Verständnis.

16. August 2011 14:16

nk funktioniert

OnlineBanking und BusinessNet

We ap

This is

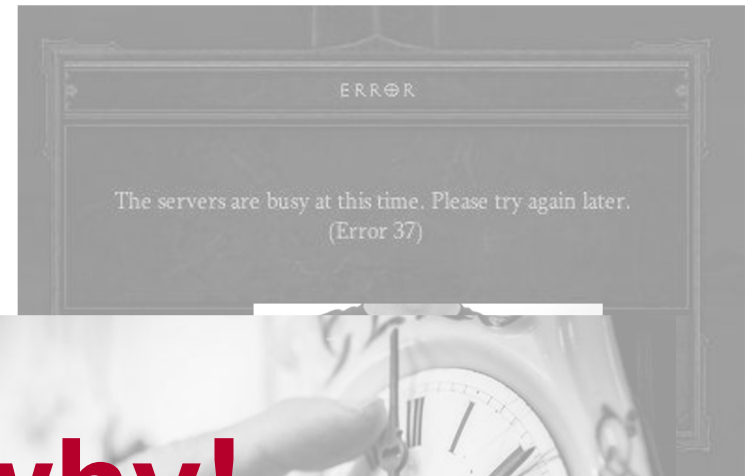
We ask

trotzdem Kur

... that's why!



Foto © APA



Kahoot!

Everything starts with requirements ...

Two types of requirements

Functional

- Defined by: customer
 - “What” is easier to define
- Easy to develop
- Easy to test and to accept
- Immediate value
- Visible to everyone ...
 - ... will be missed at once

Non Functional

- Defined by: customer!? They should
 - “How” is hard to define
- (Often) Not developed by purpose
- More difficult to test
- Not valued (enough), long term value
- Not visible at once ...
 - ... often just before/after Going-Live or under specific circumstances (eg. high load)

Two types of requirements

Functional

- Defined by: customer
 - “What” is easier to define
- Easy to develop
- Easy to test and accept
- Immediate value
- Visible to everyone ...
 - ... not missed at once

Always specified!

Non Functional

- Defined by: customer!? ? could
- “How” is hard to define
- (Often) Not developed for purpose
- More difficult
- Not valuable (high), long term value
- Not visible once ...
 - ... often lost before/after Going-Live or under specific circumstances

Often disregarded!

„*ility“ Requirement Types

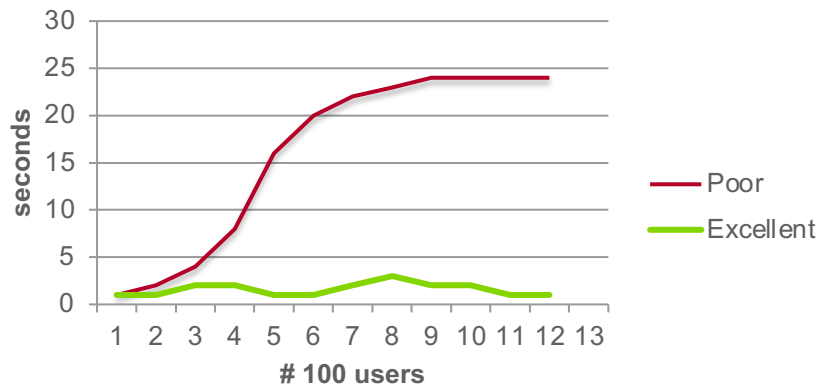
- Performance
- Usability
- Maintainability
- Scalability
- Availability
- Extensibility
- Security
- Portability

What if NFR are forgotten?

High Backend Load



Bad Response Times



Bad UX



Bad Scalability

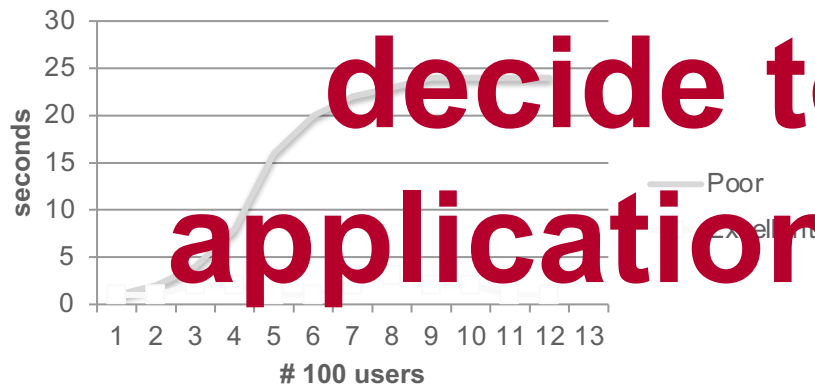


What if NFR are forgotten?

High Backend Load

```
1 [|||||] 100.0%
2 [|||||] 100.0%
3 [|||||] 100.0%
4 [|||||] 100.0%
Mem[|||||] 5.03G/7.79G
Swp[|||||] 9.04M/3.49G
```

Bad Response Time



Bad UX



Bad Scalability



...that's when you (often)
decide to build your
application from scratch!

(some other) Challenges of Efficiency Testing

Testing != Testing: Functional vs. Efficiency Testing

Functional Testing

- By hand or tool
- (Usually) On GUI level
 - ... or sometimes using the API
- (Almost) None parallel testing needed/done
- All kind of tests
 - Positive/negative, boundary, ...
 - “Everything” is tested
- Test hardware and synthetic data sufficient

Efficiency Testing

- (Almost always) With tools
- (Almost) Never via GUI
- Always using many parallel (virtual) users
- Only positive tests
 - The “happy” path
 - Only most important UseCases
- Production like hardware and data necessary

Testing != Testing: Functional vs. Efficiency Testing

Functional Testing

- By hand or tool
- (Usually) On the model
 - ... or sometimes using the API
- (Almost) always parallel tests needed/done
- A lot of tests
 - Positive/negative, boundary, ...
 - “Everything” is tested
- Test hardware and synthetic data sufficient

Easier to do!

Efficiency Testing

- (Almost always) With tool
- (Almost) Never parallel
- Always on production (virtual)
- Only a few tests
 - The “happy path”
 - Only most important UseCases
- Production like hardware and data necessary

Very technical, more challenging!

The (underestimated) process of a LPT „Project“

The (underestimated) process of a LPT „Project“



Load Test
Professional



Test/Project-
manager

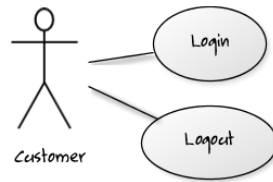
The (underestimated) process of a LPT „Project“

Working, stable
software & functionality

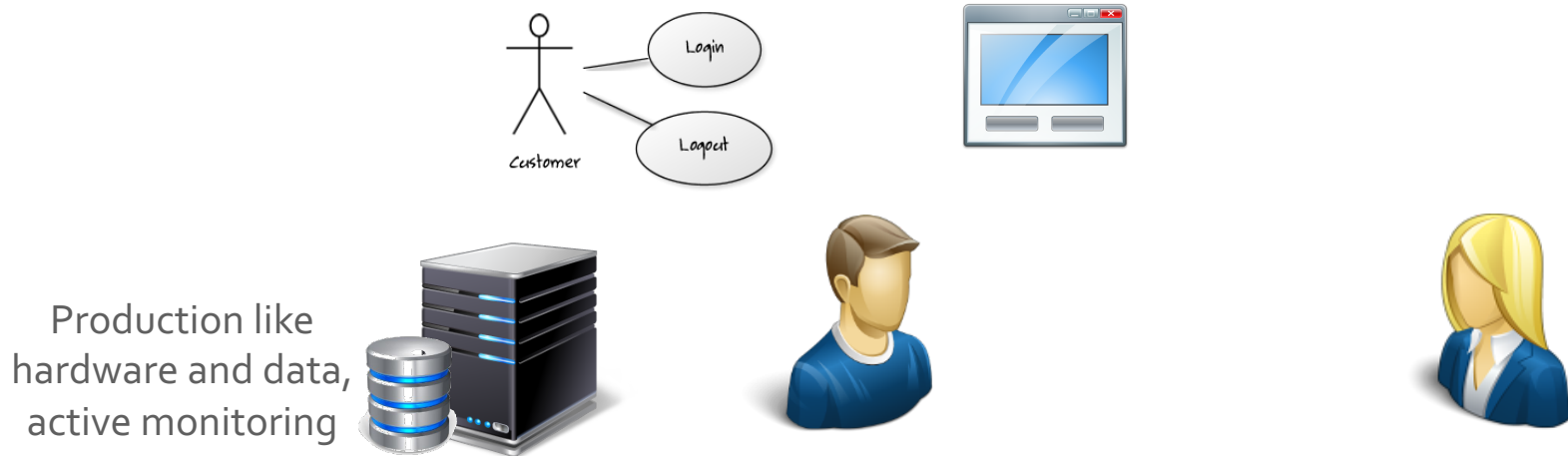


The (underestimated) process of a LPT „Project“

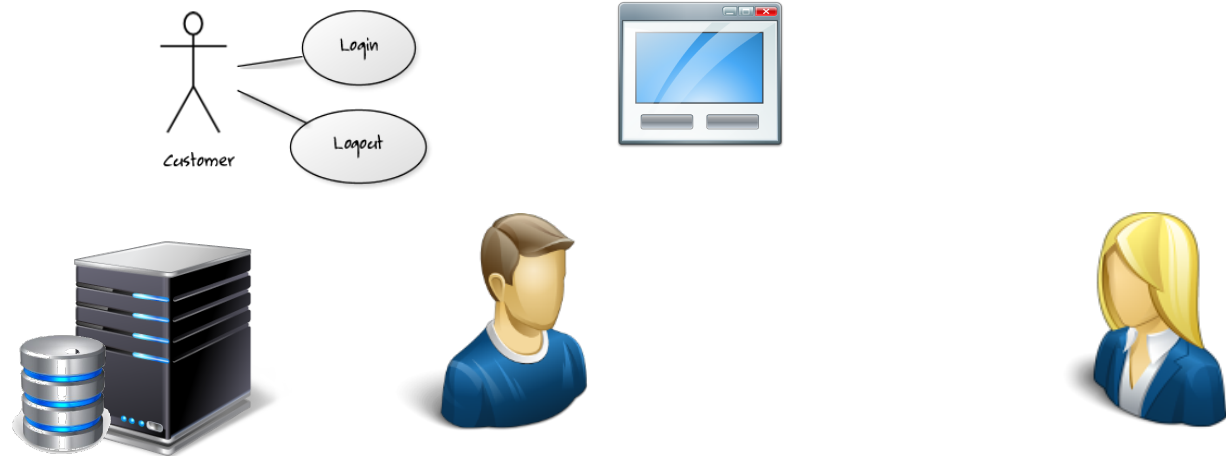
The most important
UseCases and their usage
profile



The (underestimated) process of a LPT „Project“

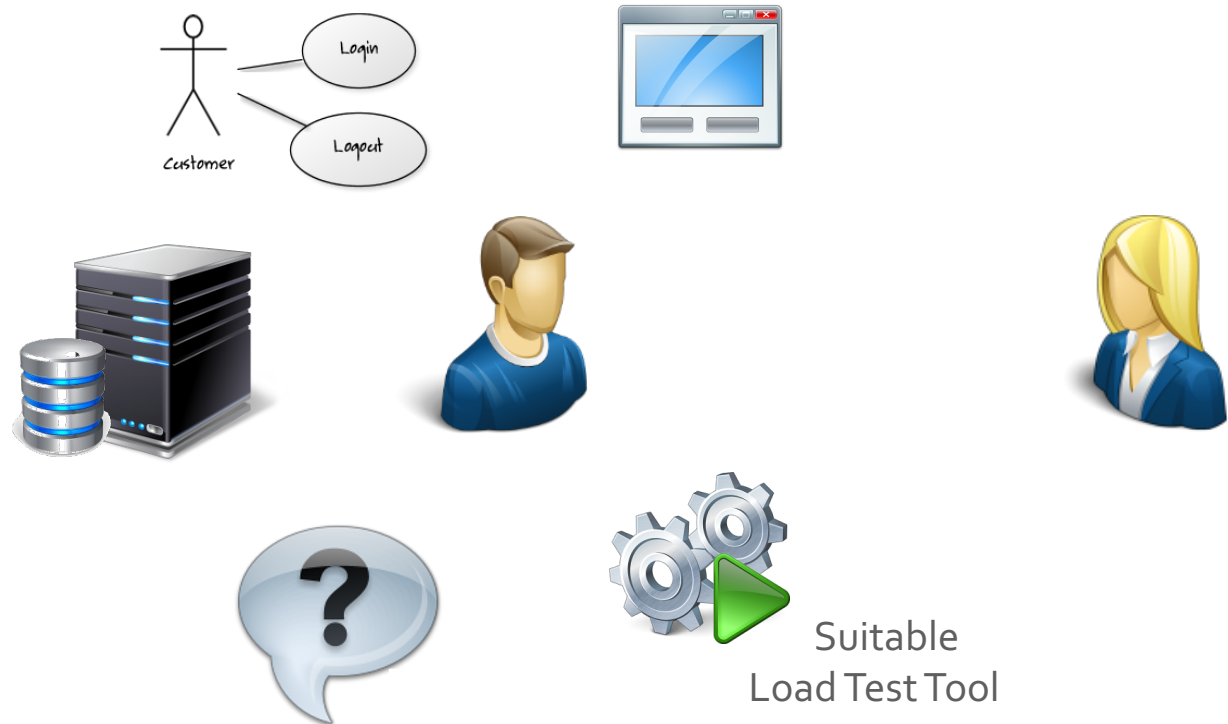


The (underestimated) process of a LPT „Project“

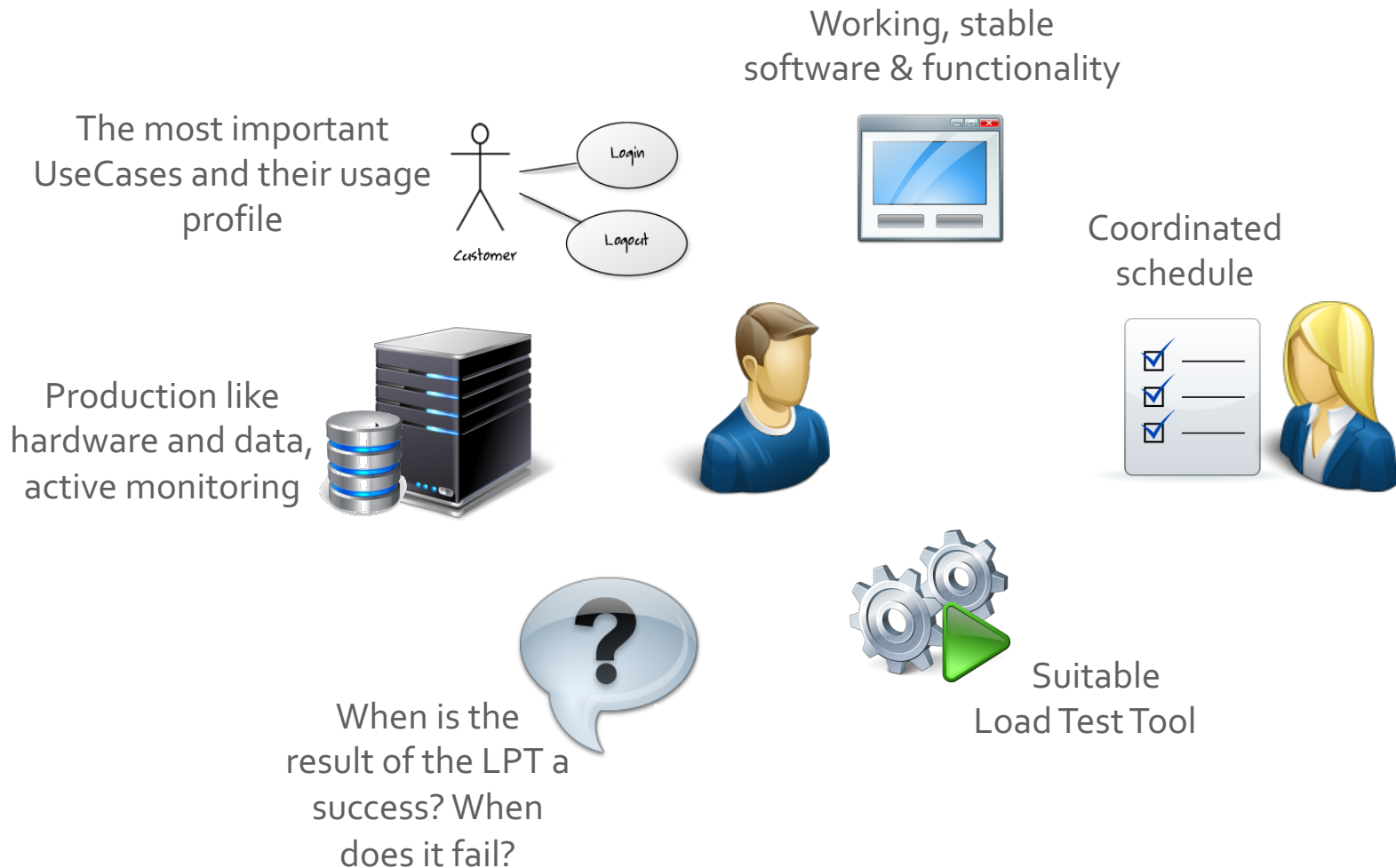


When is the
result of the LPT a
success? When
does it fail?

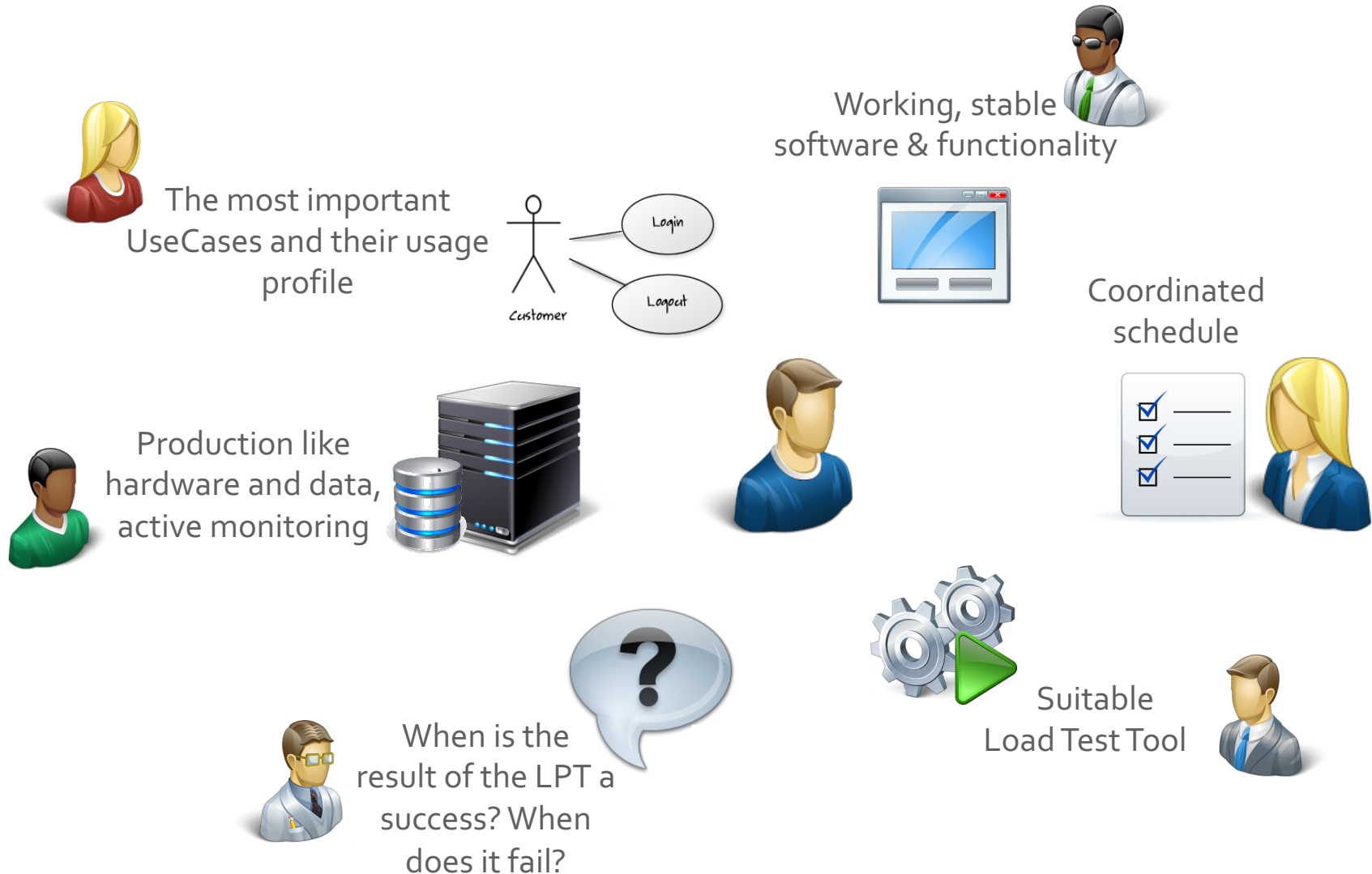
The (underestimated) process of a LPT „Project“



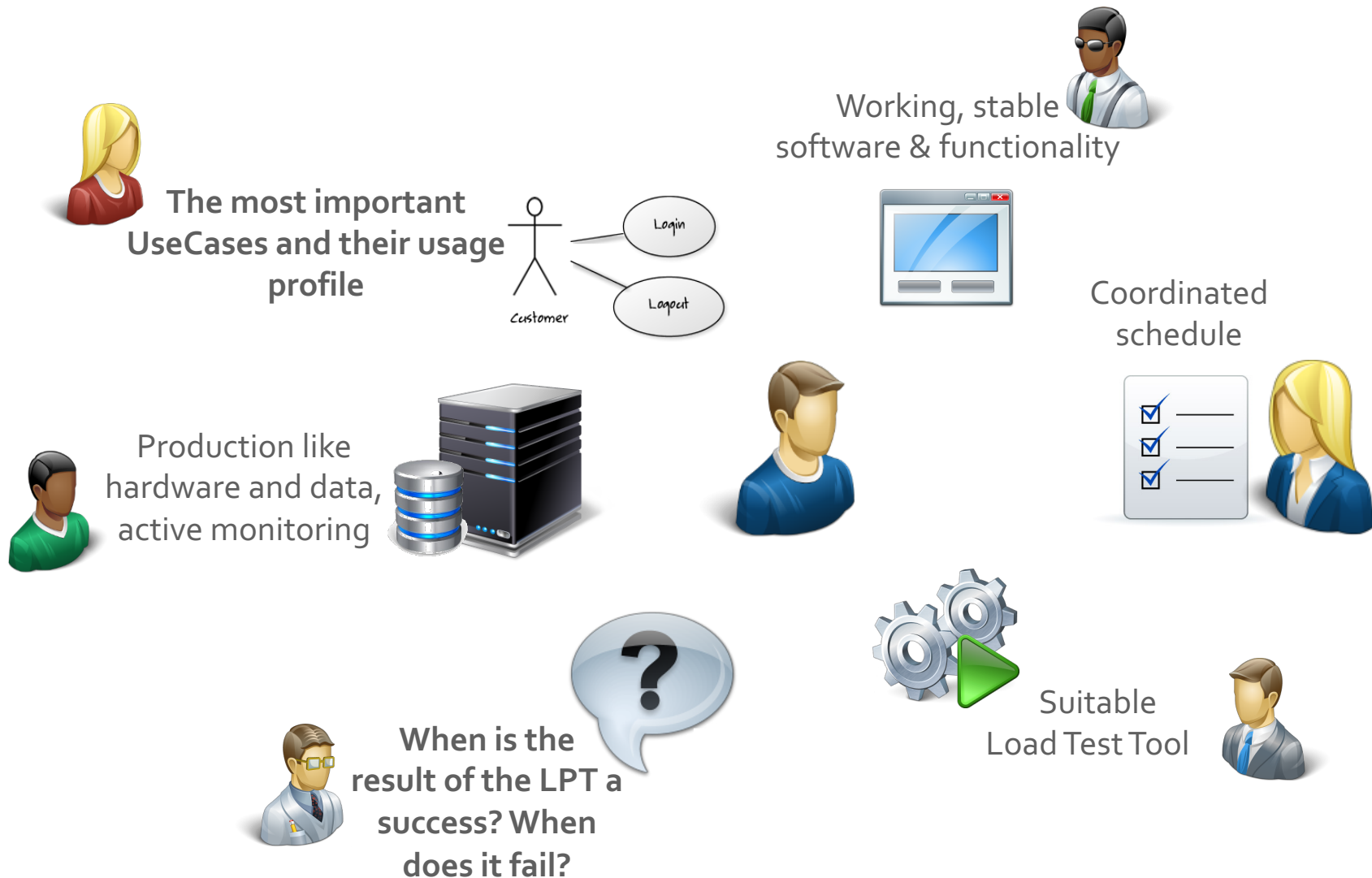
The (underestimated) process of a LPT „Project“



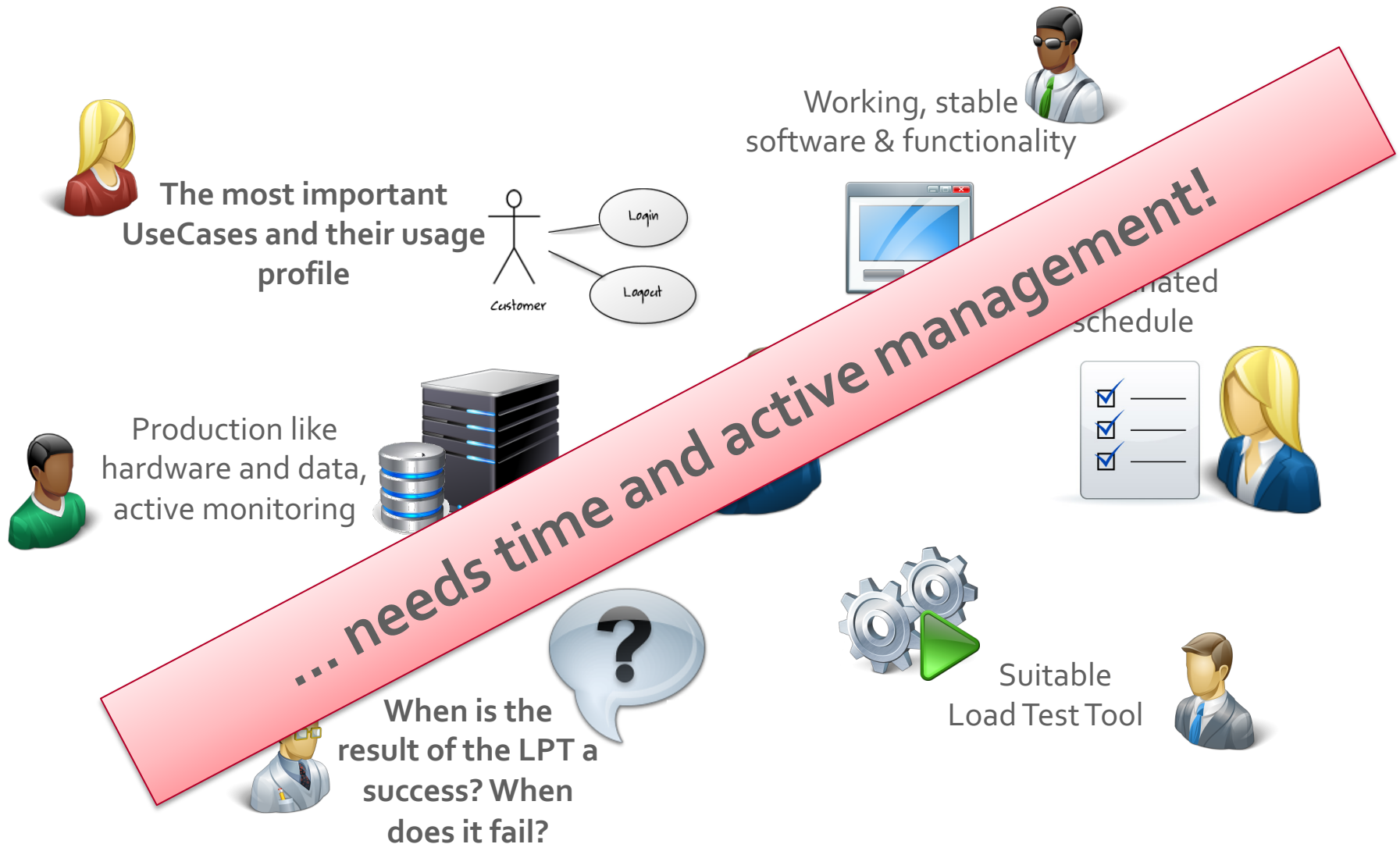
The (underestimated) process of a LPT „Project“



The (underestimated) process of a LPT „Project“



The (underestimated) process of a LPT „Project“



Kahoot!

(Trying to) Simulate reality, but ...

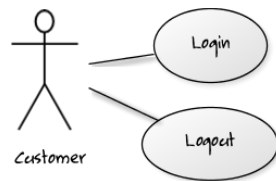
(Trying to) Simulate reality, but ...

Application &
environment under Test
→ different to PROD

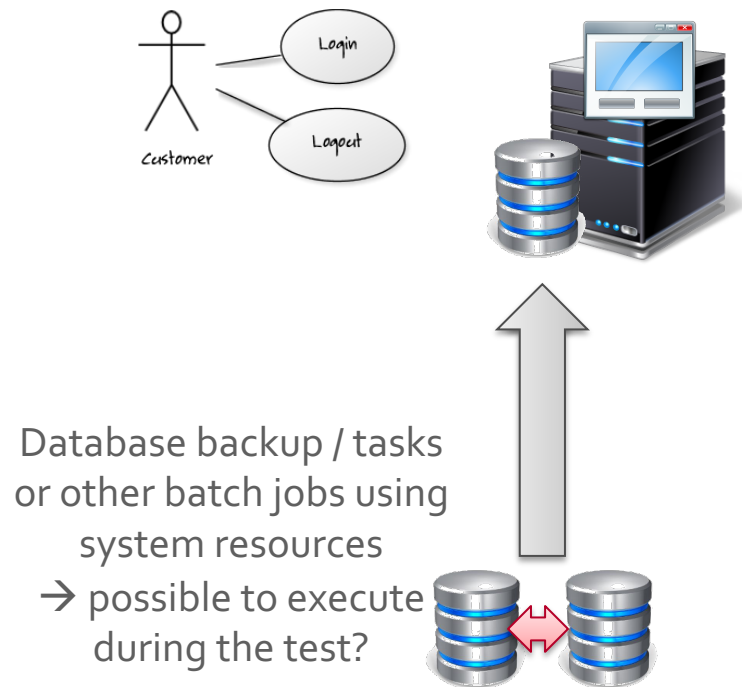


(Trying to) Simulate reality, but ...

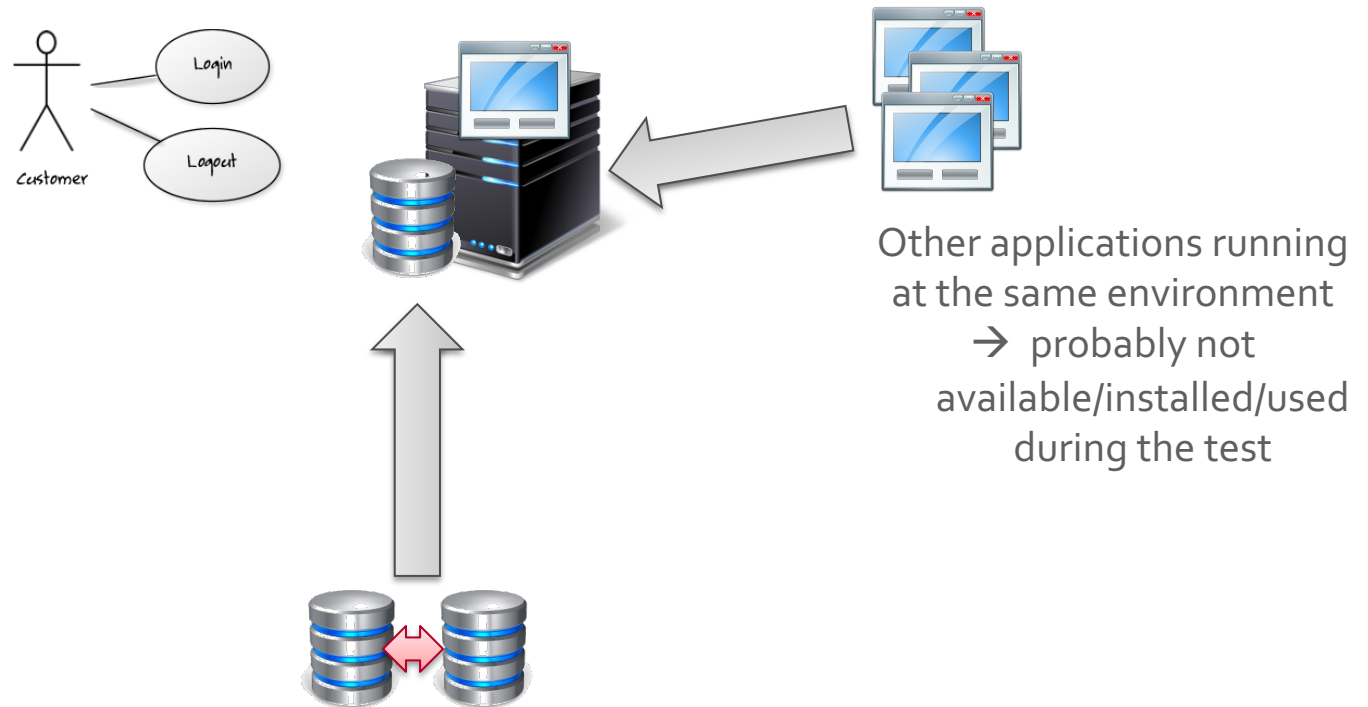
Many UseCases and even
more functionality
→ more than we can simulate



(Trying to) Simulate reality, but ...

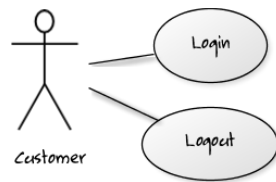


(Trying to) Simulate reality, but ...



(Trying to) Simulate reality, but ...

Many UseCases and even more functionality
→ more than we can simulate




Application & environment under Test
→ different to PROD



Database

Can we simulate



Customer

Login

Logout

Database or...

...ute

...e test?

100% reality is not possible!

Pareto-Principle (80/20) and risk assessment

...er applications runn...

at the same environme...

→ probably not

available/installed/u...

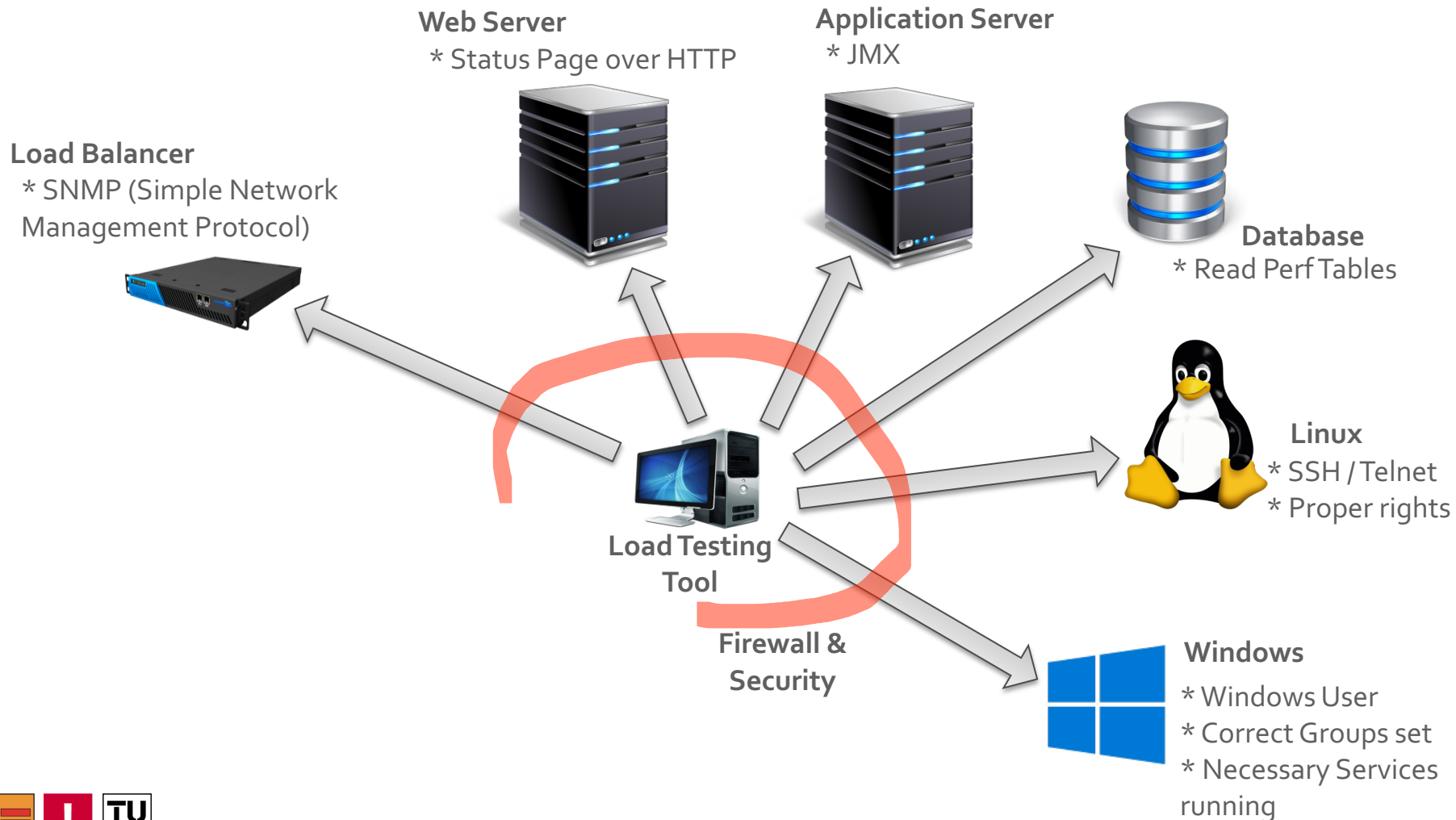
during the test

→ probably not available/installed/used during the test

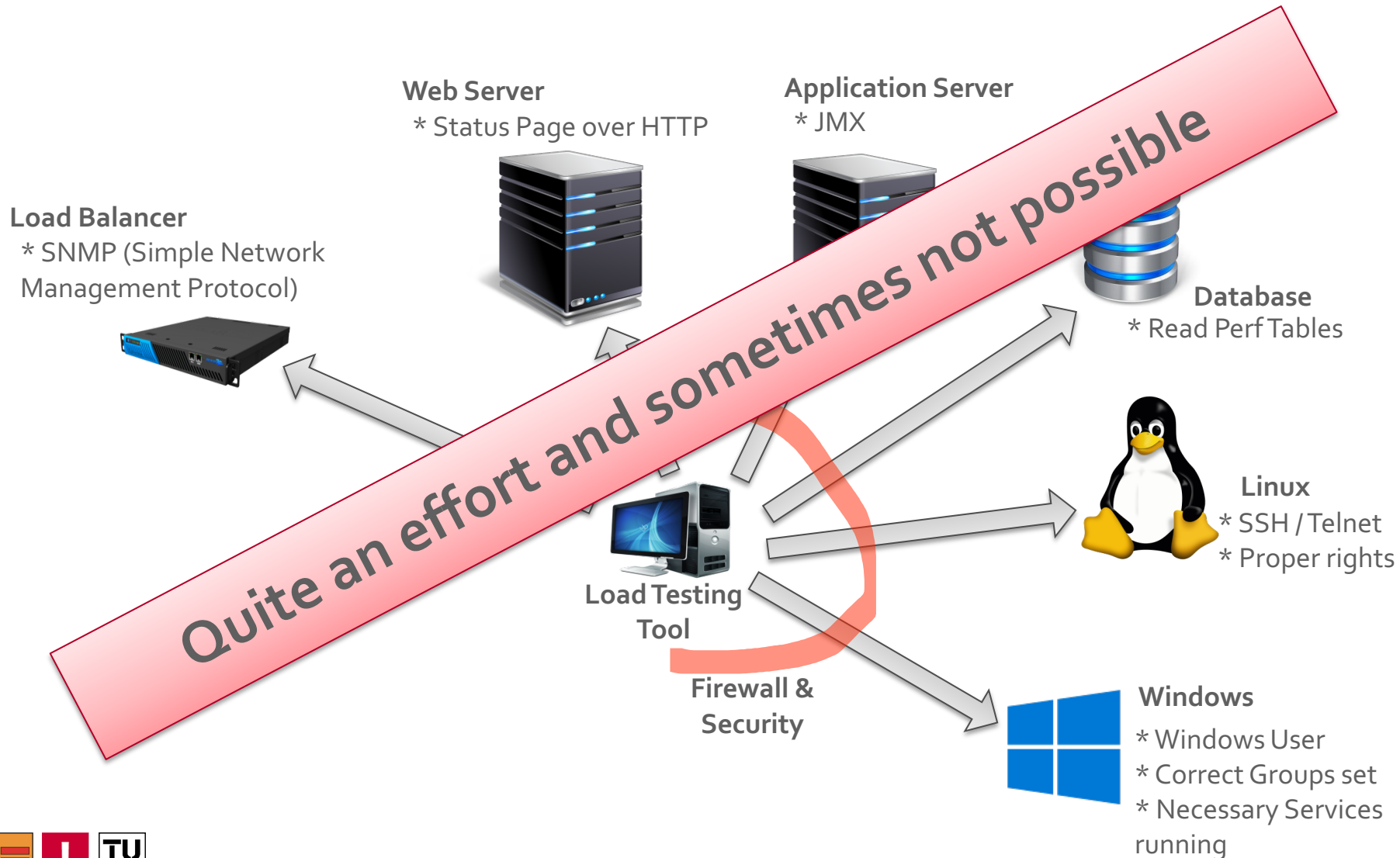


Monitoring

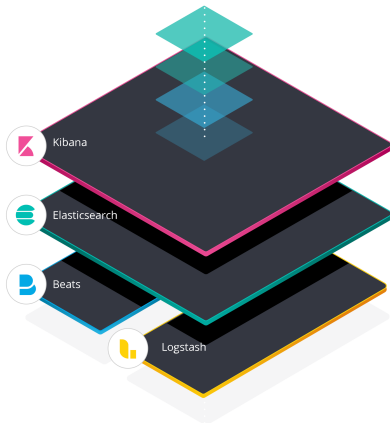
Monitoring: using the Load Testing Tool



Monitoring: using the Load Testing Tool



Monitoring: using Company Tools

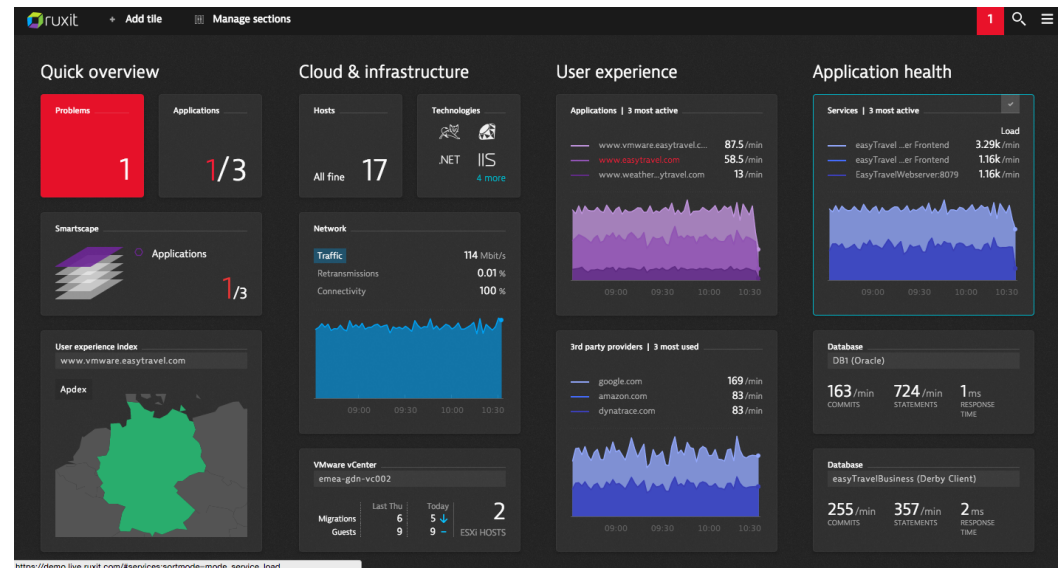


dynatrace

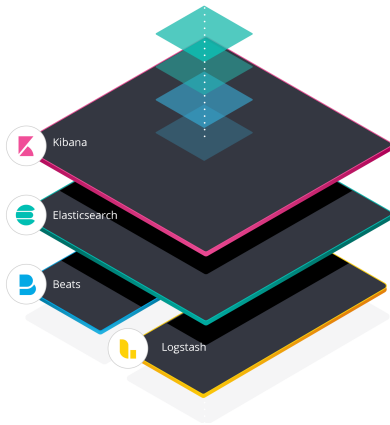
Nagios®



Software Testing



Monitoring: using Company Tools

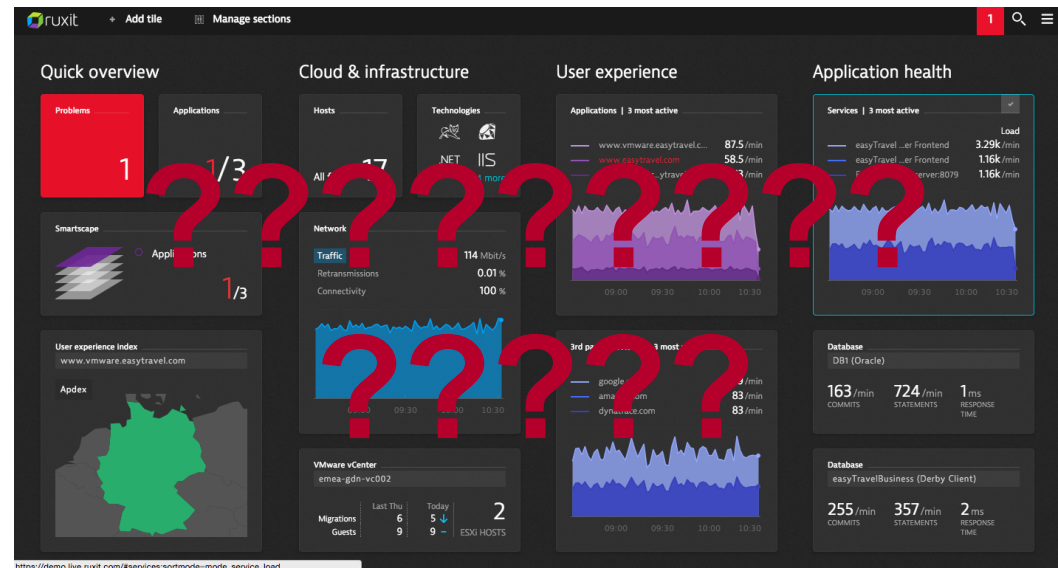


dynatrace

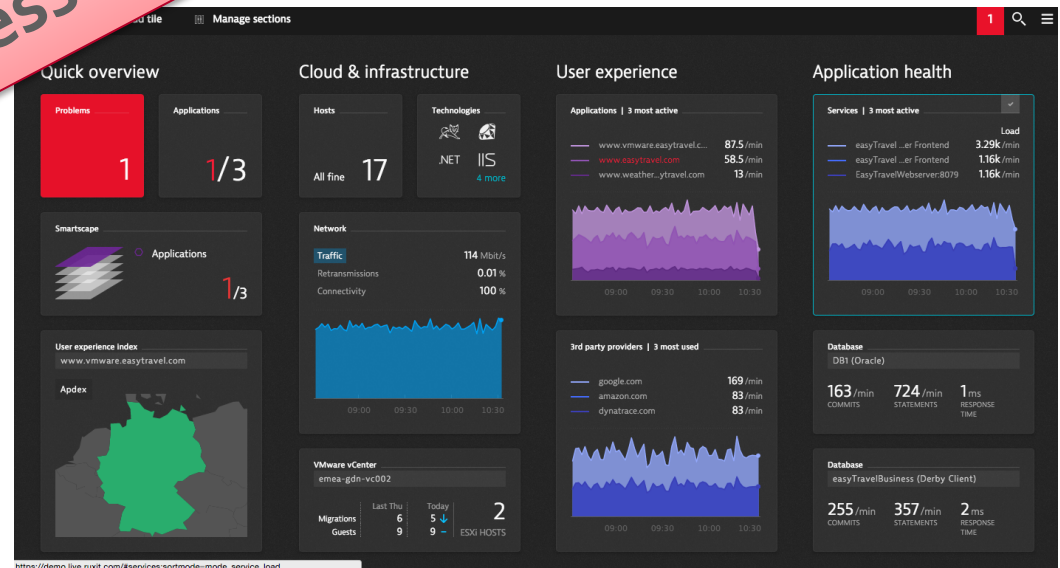
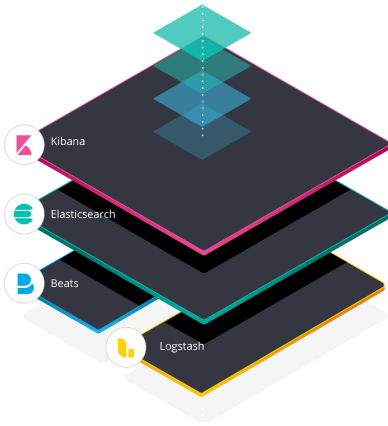
Nagios®



Software Testing



Monitoring: using Company Tools



Kahoot!

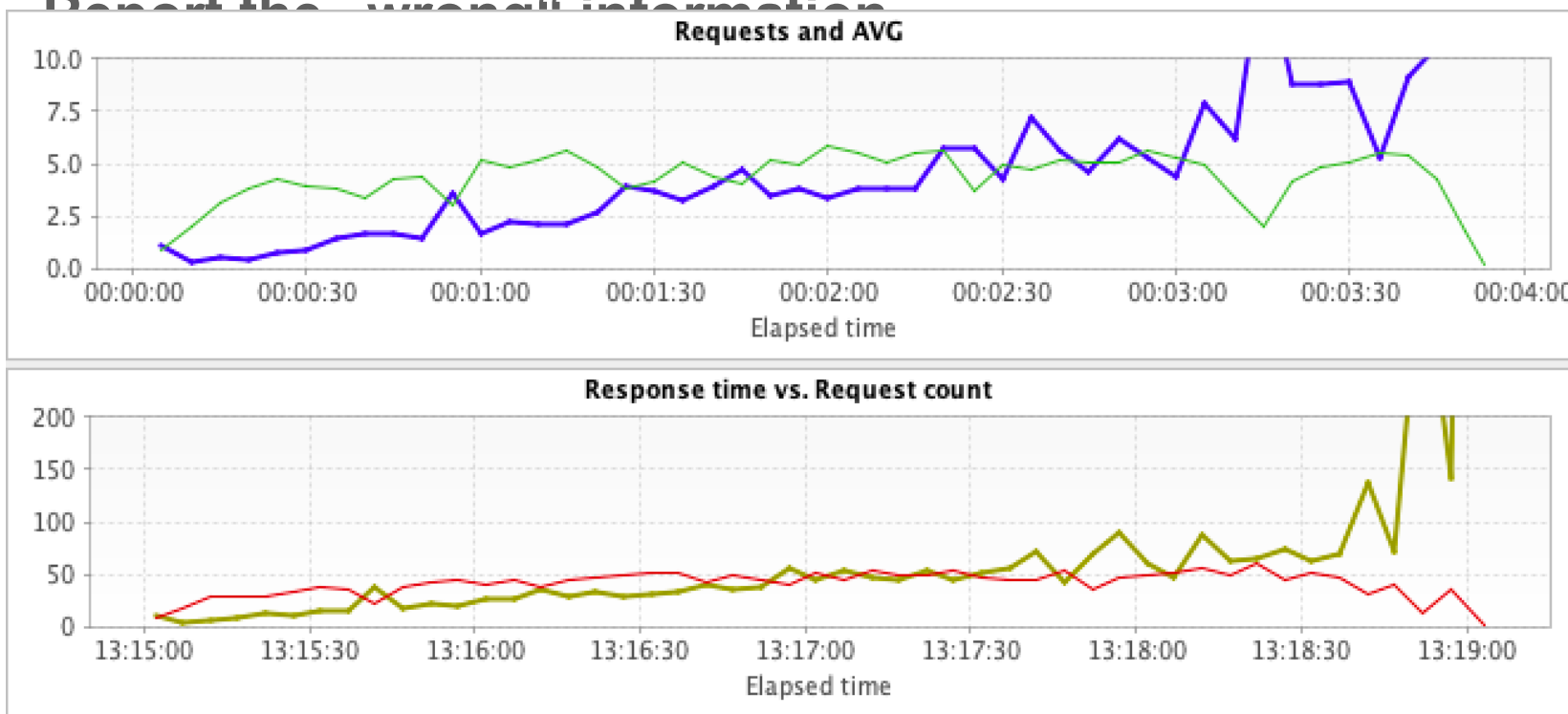
Reporting: Pitfalls

- **Ambiguous statements**
 - x-times or xxx% faster/slower
 - ... does not mean anything without setting the context
- **Report the „wrong“ information**
 - Focus your stakeholders
 - Talk „business“ and/or „technical“
- **Only analyze and report average numbers**
 - Always have a look at min, max, deviation, percentiles
 - The average lacks a lot information
- **Difficult to read and compare graphs**
 - If they need comprehensive description, the are probably not good
 - Stay consistent as much as possible

Reporting: Pitfalls

- **Ambiguous statements**
 - x-times or xxx% faster/slower
 - ... does not mean anything without setting the context

- **Report the wrong information**



od

Bottom line ...

- **Challenging in different ways**
- **Wide skillset necessary**
 - Technical
 - Analytical
 - Communication
- **... but this makes it also very interesting**

Kahoot!

The technical approach

Manual Load Testing (Crowd Testing): Friday Night PP

*“We need a load test, we don’t have tools
but we have many employees - let’s do it!”*

Advantages

- No further purchases needed (ok, beer & pizza)
- Realistic user load
- No (test) “experts” needed
- Getting the customers involved!

Disadvantages

- Not repeatable
- Employee time is expensive too!
- Test coordination and preparation difficult
- Difficult to collect and analyse the (subjective) results



Manual Load Testing (Crowd Testing): Friday Night PP

*“We need a load test, we don’t have tools
but we have many employees - let’s do it!”*

Advantages

- No further purchases needed (ok, beer & pizza)
- Realistic user load
- No (test) “experts” needed
- Getting the customers involved

Disadvantages

- Not repeatable
- Employee based - expensive tool!
- Test design and preparation difficult
- Hard to collect and analyse the (subjective) results

Only sometimes a good idea!



Load Testing using GUI-Automation Tools

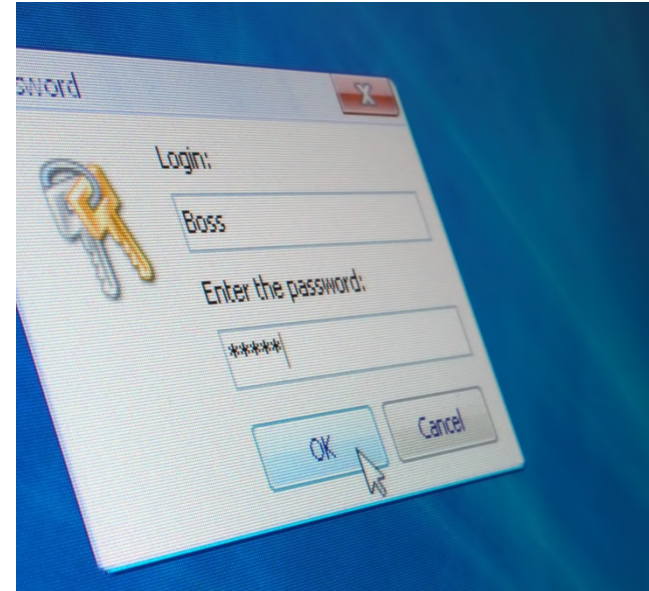
“We already have automated tests, let’s use them!”

Advantages

- Repeatable

Disadvantages

- GUI-Automation is (often) unreliable
- Difficult to control, not scalable
- Hardware and license cost (if commercial)
- Difficult to collect the results



Load Testing using GUI-Automation Tools

“We already have automated tests, let’s use them!”

Advantages

- Repeatable

Disadvantages

- GUI-Automation is (often) slow
- Difficult to control
- Hardware requirements (if commercial)
- Difficult to get results

(Almost) never a good idea!



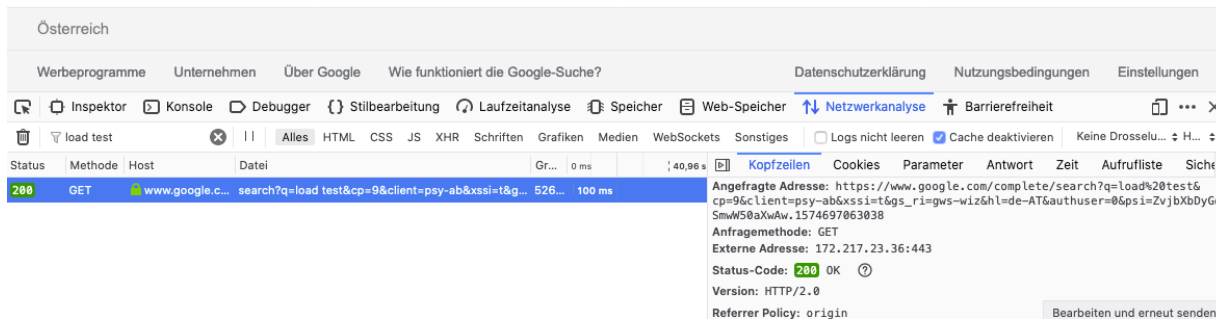
Solution: Load Testing using the protocol tier



Google

load test

GET https://www.google.at/search?q=load+test



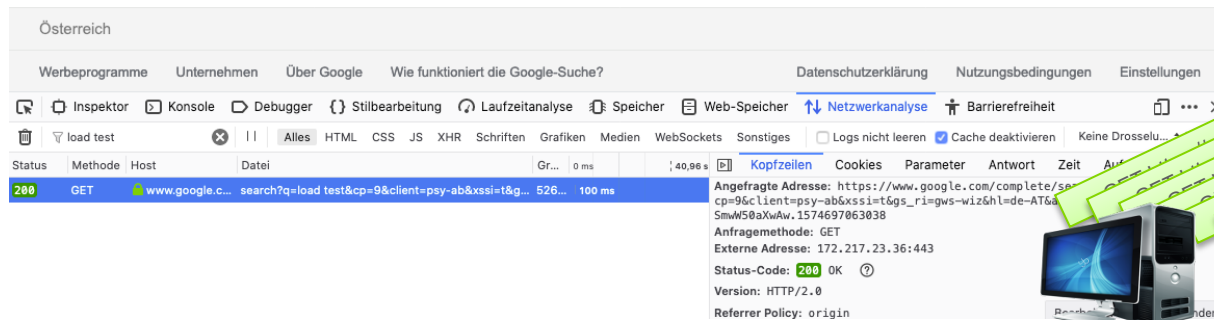
Solution: Load Testing using the protocol tier



Google

load test

GET https://www.google.at/search?q=load+test



GET https://www.google.at/...



Using Load Test Tools and the protocol tier

“Let’s do a load test, the right way!”

Advantages

- Repeatable, scalable and stable
- Result collection and analysis
 - when using the right tools
- Headless (no client involved)
 - Less resources, simulation of many (virtual) users on one machine possible
 - **BUT:** no client means you have to take care of a lot of things yourself (**no Browser, no JS**)



Disadvantages

- Much more complex than other test practices
- Licence cost for commercial Load Test Tools
- No client involved
 - → no client performance measured
 - → lack of trust in the results

Using Load Test Tools and the protocol tier

“Let’s do a load test, the right way!”

Advantages

- Repeatable, scalable and stable
- Result collection and analysis
 - when using the right tools
- Headless (no client involved)
 - Less resources, simulation of many users on one machine possible
 - **BUT:** no client means you have to take care of a lot of things yourself (**no Browser, no JS**)

Disadvantages

- Much more complex than other test practices
- Licenses for commercial Load Test Tools
- Not always used
 - Only server performance measured
 - → lack of trust in the results

Correct approach!



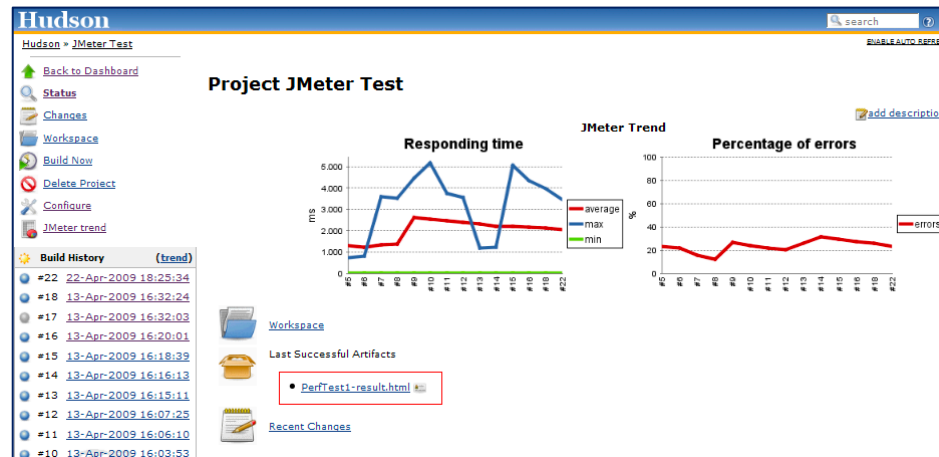
Kahoot!

When to execute Efficiency Tests?

- Before Going-Live (of course)
- When changing hardware
- When changing essential frameworks/software
- When changing (infrastructure/application) configuration
- To simulate the future

On every Build!?

- Trend analysis!



Efficiency Testing in DevOps?

- **From code change to production in hours**
 - **Much more automation needed**
 - Load Tests on every build
 - **Monitoring and analysis of the production environment**
 - Key Performance Indicators (business & technical) alerting
 - Important information for your next Load Tests
- **Performance Engineering!**



Kahoot!

The tools

(Some) Load Testing Tools

Open Source

- Apache JMeter
- Gatling
- Grinder
- OpenSTA
- ...

Commercial

- Neotys NeoLoad
- Micro Focus (HP) LoadRunner
- Micro Focus Silkperformer
- Radview
- IBM Rational Performance Tester
- Visual Studio Load Test
- SmartBear ReadyAPI / LoadComplete / LoadNinja
- Octoperf
- ...

- **Long term competitor**

- NeoLoad 1.0: 2005

- ~~**Free 50 Virtual User license**~~

- Since February not available any more!

- **License for 1000 Virtual Users**

- 4 out of 365 days rental: ~2,6k€
 - 1 year rental: ~26k€

- **NeoLoad Web**

- slow transition to web “IDE”

Advantages

- Compatible with Linux/Win/macOS
- No scripting needed
 - Point’n’Click “programming”
- Very good monitoring capabilities
- “Cheap”
- Very Good Support
- Agile Product Development
- Full focused

Disadvantages

- (Almost) Only Web technologies
- No scripting possible

Common tool approaches

Recording

- Usually done via HTTP Proxy (for browser applications)
- Helps to quickly build a rough draft of the UseCase
- Necessary? No, but speeds up the whole scripting process

Request/Response correlation

“Use something from a previous response in a following request” (e.g. Session information)

→ Remember: there is no Browser (tools use the protocol tier), so no there is JavaScript execution

Login with Username & Password



Extract SessionID from the response: SESS_ID=ADFJDAADFISDANF



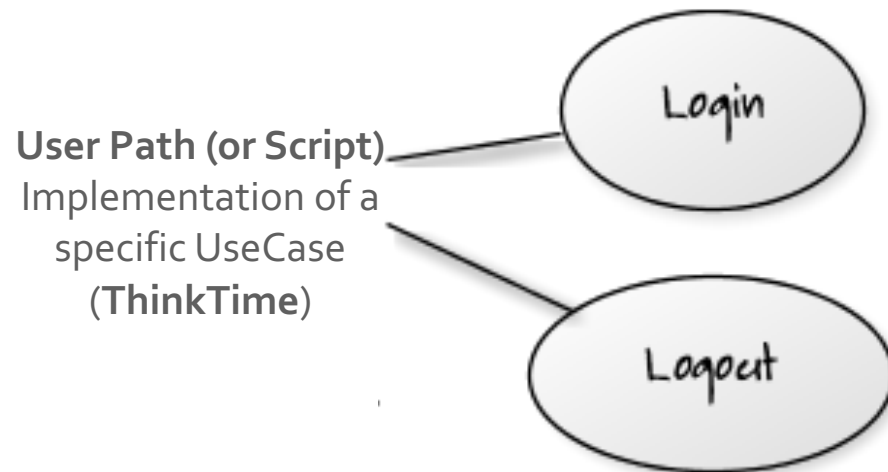
Send next request using the extracted SessionID: SESS_ID=ADFJDAADFISDANF



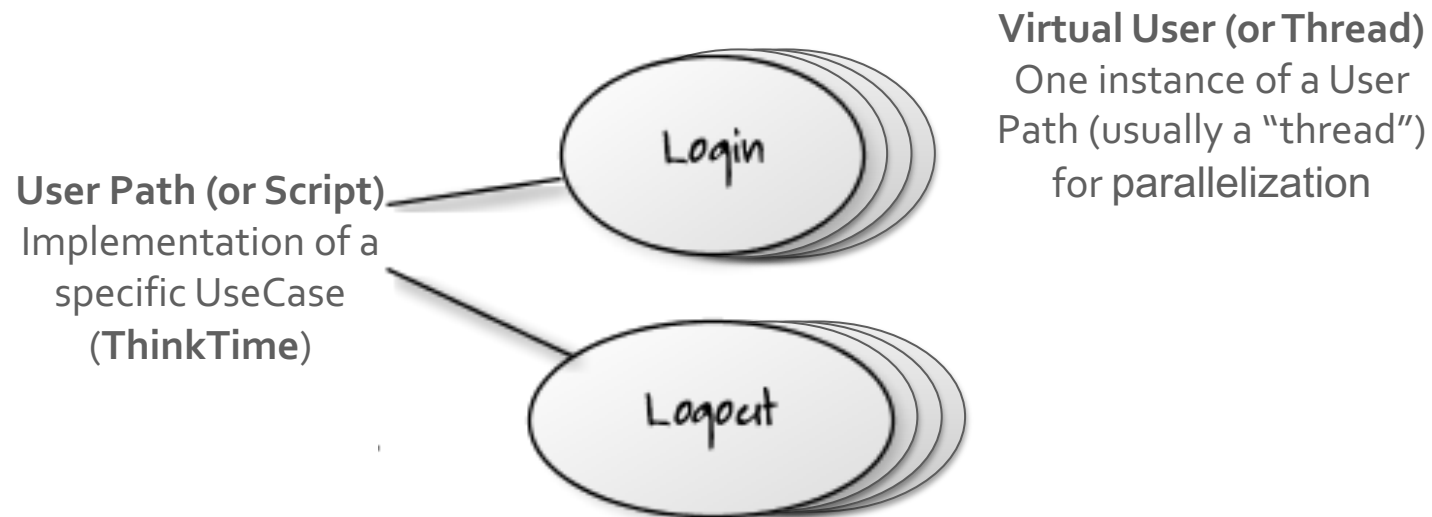
„OK“



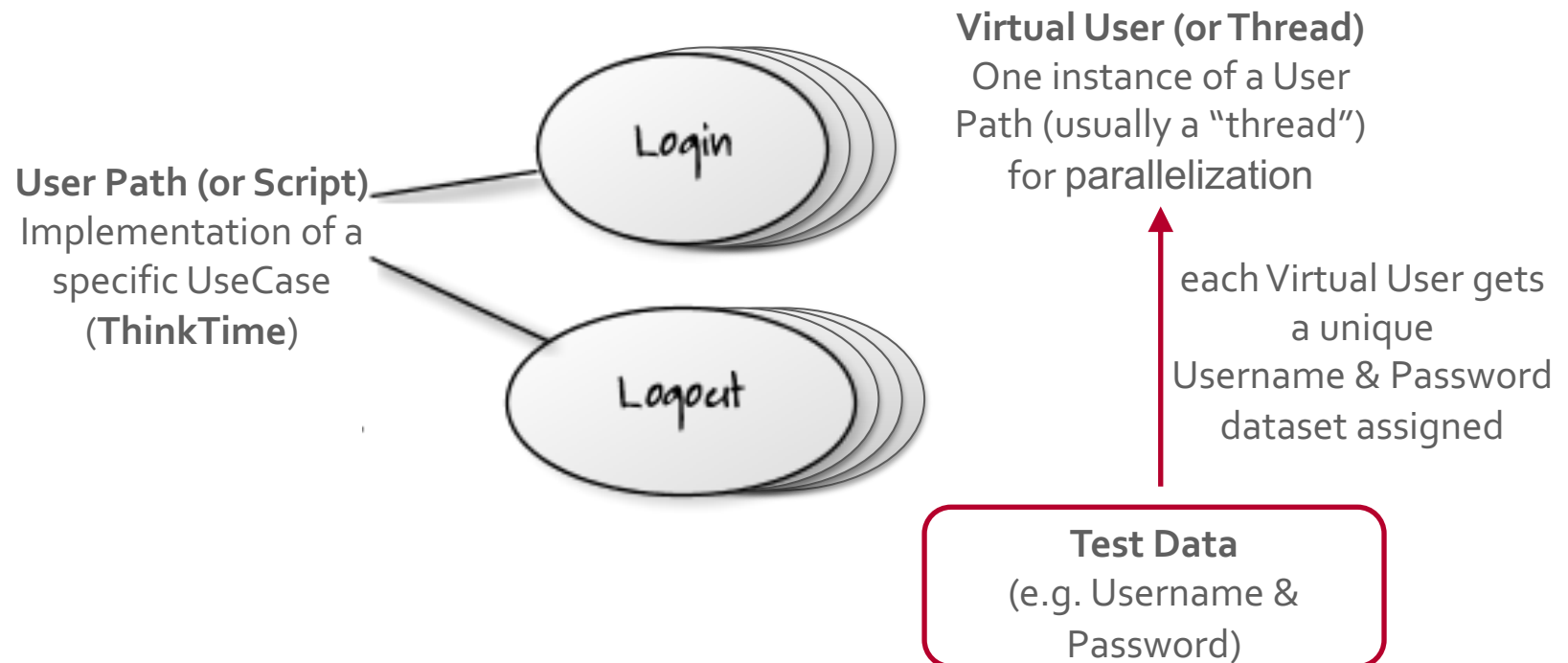
User Path, Virtual User, Test Data, Scenario



User Path, Virtual User, Test Data, Scenario



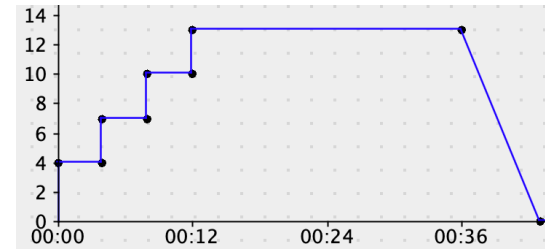
User Path, Virtual User, Test Data, Scenario



User Path, Virtual User, Test Data, Scenario

Scenario

Combination of different User Paths and amount of Virtual Users (load policy) per User Path



User Path (or Script)
Implementation of a
specific UseCase
(ThinkTime)



Virtual User (or Thread)

One instance of a User
Path (usually a "thread")
for parallelization

each Virtual User gets
a unique
Username & Password
dataset assigned

Test Data
(e.g. Username &
Password)

Demo Setup

Jira UseCase

- Log into Jira
- Create a new Issue
- Open the created Issue
- Logout



NeoLoad

- Record the UseCase
- Manually correlate (XSRF and ATL Token) and enhance recording with Test Data
- Run Load Test, monitor response times, basic nginx and Linux KPIs

