



Ten more **things**

Performance Engineering: „Was wäre, wenn ... ?“

Klemens Loschy

Warum überhaupt PE?

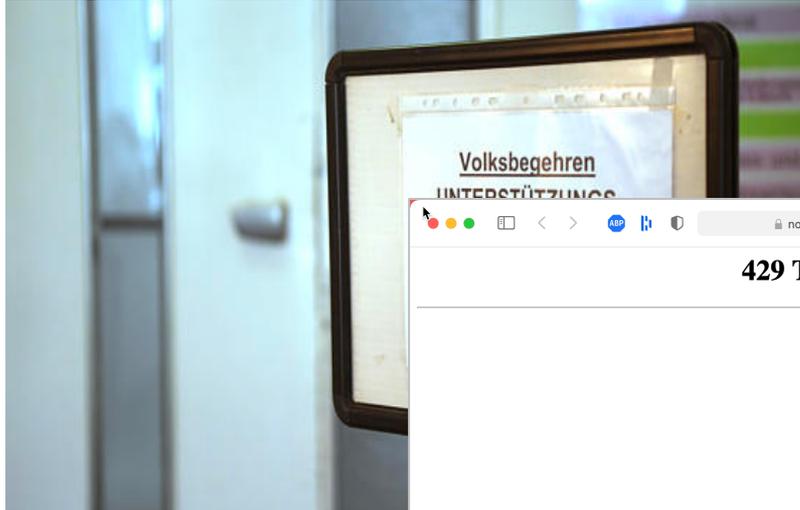


„Don't Smoke-Volksbegehren:
Erneut IT-Probleme bei
Unterstützungserklärungen“
(Die Presse, 20.02.2018)

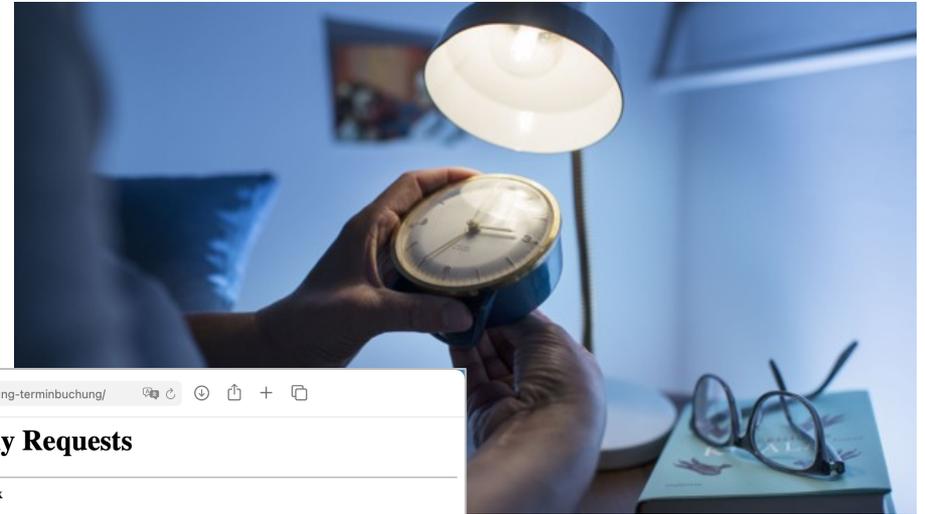


„Sommerzeit abschaffen? Umfrage-Server
kollabiert“
(Die Krone, 06.07.2018)

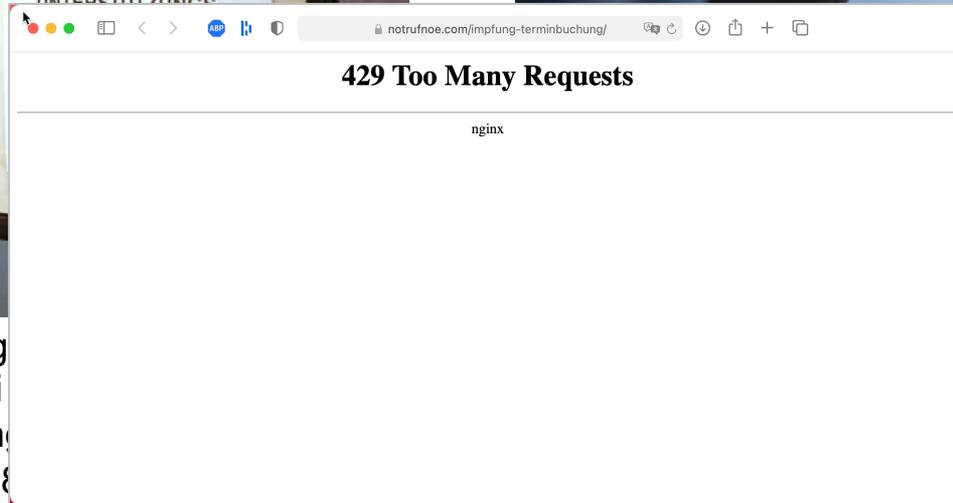
Warum überhaupt PE?



„Don't Smoke-Volksbeg
Erneut IT-Probleme bei
Unterstützungserklärung
(Die Presse, 20.02.2018)

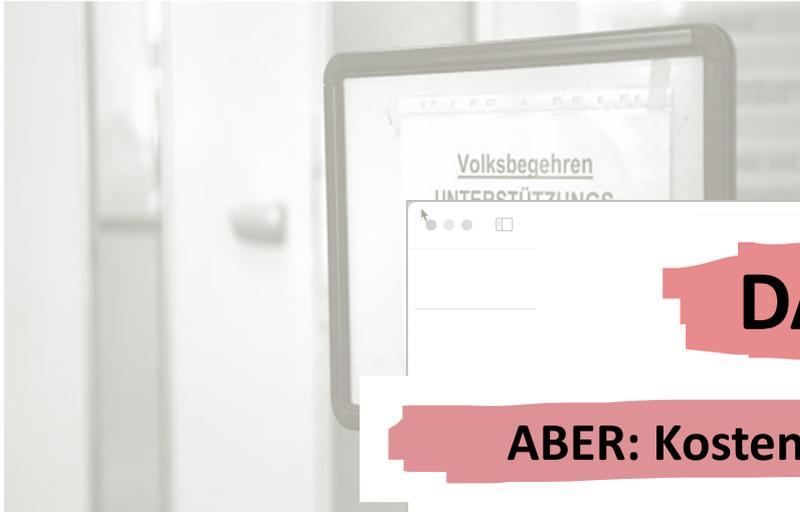


Umfrage-Server



Corona Impfanmeldung auf Notruf NOE (10.02.2021)

Warum überhaupt PE?

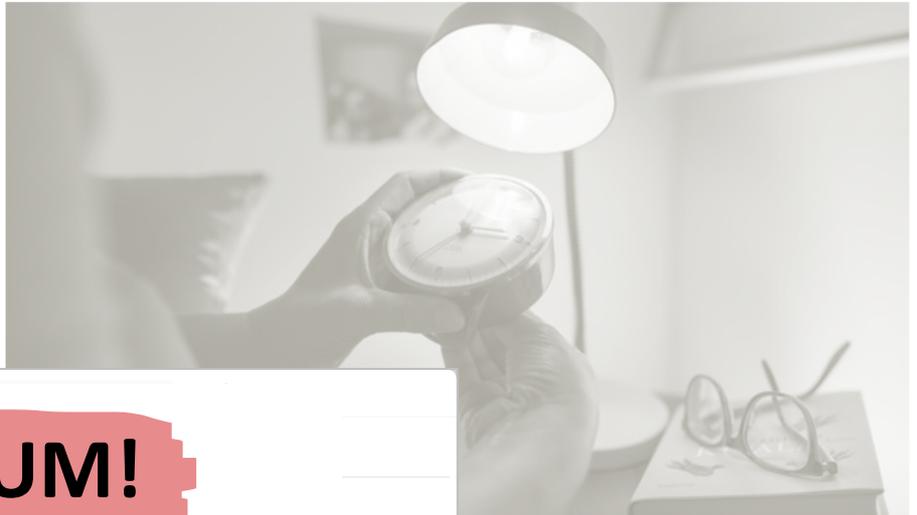


DARUM!

ABER: Kosten vs. Nutzen abwägen!

„Don't Smoke-Volksbeg
Erneut IT-Probleme bei
Unterstützungserklärung
(Die Presse, 20.02.2018)

Corona Impfanmeldung auf Notruf NOE (10.02.2021)



Umfrage-Server

Die Folgen eines Ausfalles

- Verunsicherung der Kunden, Imageverlust
 - Damit Abwanderung zum Mitbewerb
- Kein „Geschäft“ während dieser Zeit
 - Dieser reale Verlust ist ganz einfach messbar
- Hohe Mitarbeiterkosten
 - Überstunden, Nacht- und Wochenendeinsätze
- Wiederanlauf der Applikation nicht möglich
 - Die Belastung wird auf Grund der Probleme kurzfristig noch steigen
- Keine Möglichkeit die root cause gesichert festzustellen
 - Und auch die potentielle Problembehebung nicht zu verifizieren

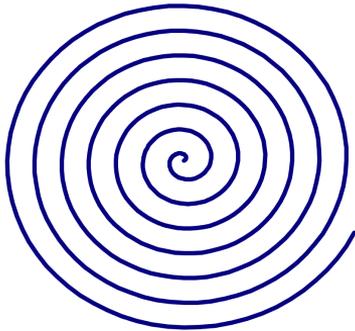


1. Bereiten Sie sich vor!

- Ausfälle werden passieren!
- Man kann sich davon „überraschen“ lassen, oder vorbereitet sein
- Planen und proben Sie den Ablauf bei einem Ausfall

Zwei einfache „Workarounds“

Soft Launch oder Rollierendes Release



Die „Spaßbremse“

A screenshot of a booking page for 'Niederösterreich Mein Land'. The page features the logo of the Austrian state of Lower Austria (Niederösterreich) at the top, which consists of a blue square with a yellow 'N' and a yellow star. Below the logo, the text reads 'Niederösterreich Mein Land'. The main heading is 'Terminbuchung für die Corona-Schutzimpfung'. Below this, a message states: 'Leider sind gerade alle unsere Buchungslinien besetzt. Sie befinden sich in der Warteschlange. Bitte haben Sie einen Augenblick Geduld. Sobald ein Zugang frei wird, werden Sie zur Buchung weitergeleitet.' A progress bar is shown below the message, consisting of a blue bar with a yellow segment and a lightning bolt icon. Below the progress bar, the text reads: 'Sie sind an der Reihe. Vielen Dank für das Warten. Sie werden nun zur Website geleitet.' At the bottom, it says 'Status zuletzt aktualisiert: 10:05:22'. There is a link 'Warteschlange verlassen (Sie werden Ihren Platz verlieren)' and a queue ID: 'Queue ID: 07d24721-df28-43da-88d1-4ce2aca8440c'.



2. Machen Sie sich nicht selbst das Leben schwer!

- Verteilen Sie wo möglich die zu erwartende Last
 - Aussendung von Werbung staffeln
 - Benutzer schrittweise umstellen
- Mittels „Spaßbremse“ die Überlast verhindern
 - Und damit gleichzeitig den Normalbetrieb absichern

Wie wird Last simuliert?

„Friday Night Pizza Party“



Für PE nicht geeignet!

Vorhande GUI Automation



Hat NUR Nachteile!

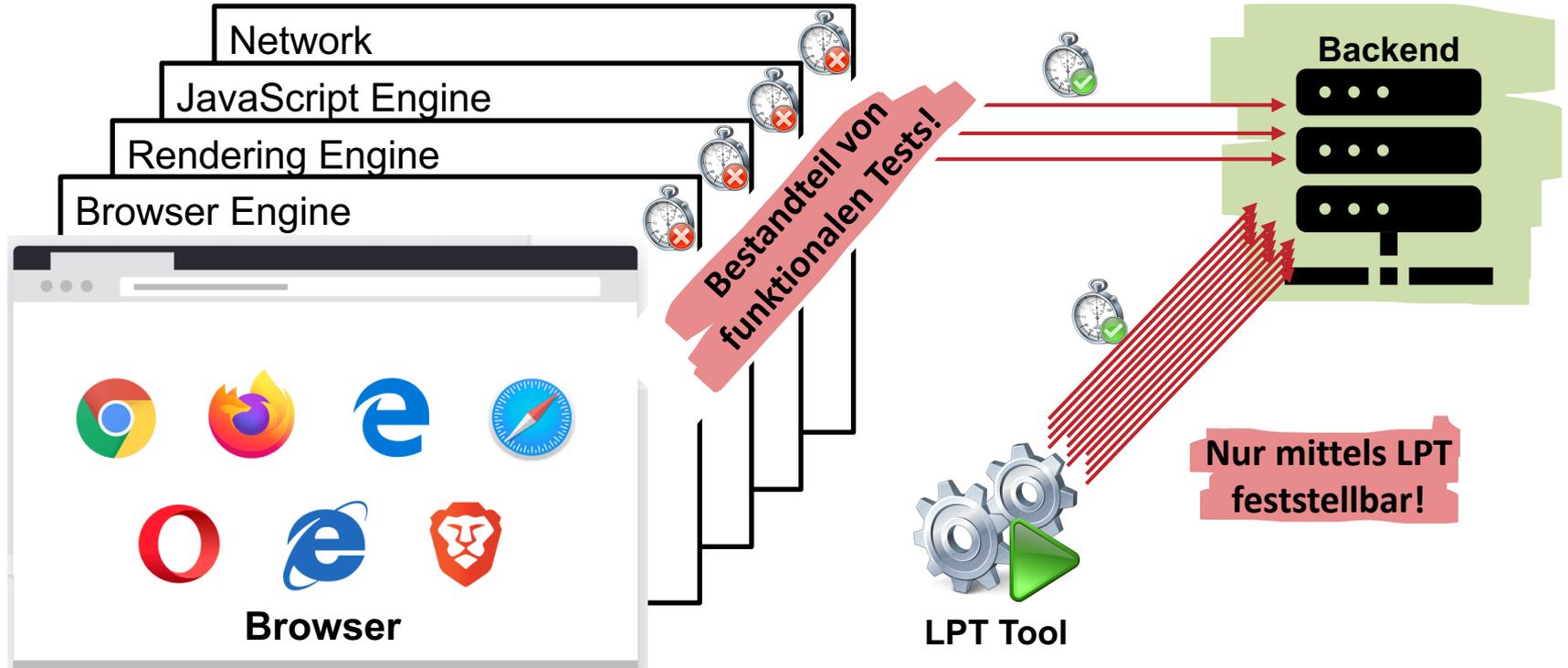
Protokoll Level



Einzig richtige Lösung!



Unterschied: User Experience und LPT





3. Verwenden Sie die richtigen Tools!

- Für Last- und Performance Tests gibt es spezielle Tools
 - Die Simulation erfolgt auf Protokoll Ebene
- Im Fokus ist ganz klar die Backend Performance
 - Die User Experience kann zusätzlich durch gezielte funktionale Tests ermittelt werden

Notwendige LPT Vorbedingungen



LPT Experte

- LPT Erfahrung bei
 - Planung
 - Durchführung
 - Analyse
- Fundiertes technisches Know How
- Kommunikation mit
 - Fachbereich
 - Entwicklung
 - Management

Notwendige LPT Vorbedingungen

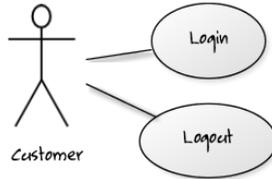
Testbare Software

- Funktional getestet
- Stabile Interfaces



LPT Experte

Notwendige LPT Vorbedingungen



Testbare Software



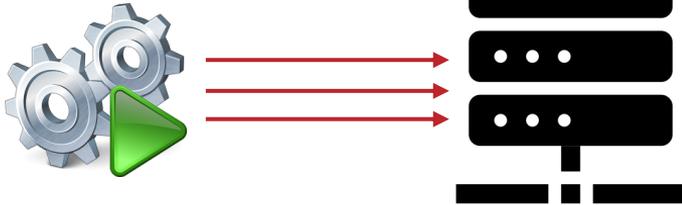
Definition der zu testenden Funktionen

- Einzelne API Calls
- Teilbereiche der Applikation
- Ganze Anwendungsfälle
- Die „Realität“
- Und: wie stark ist die jeweilige Belastung



LPT Experte

Wieso Anwendungsfälle simulieren?



Kein direkter Einfluss
der Anwendungsfälle
auf das Backend

ABER: Was wäre wenn?

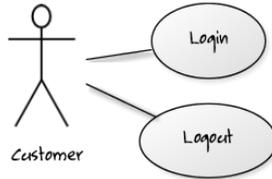
- Unsere Kunden mehr die Handy App anstatt die Web App verwenden?
- Wir den Ablauf der Applikation anpassen?
- Wir auf jeder Seite zusätzlich einen Tracker einbauen?
- Durch verstärkte Marketingmaßnahme der Onboarding Prozess vermehrt durchgeführt wird?
- → **Fachliche Fragestellung sind nur durch die Simulation von Anwendungsfällen zu beantworten!**



4. Mit Anwendungsfällen die Zukunft simulieren!

- Anwendungsfälle können beliebig miteinander kombiniert werden
- Alle denkbaren (und undenkbaren) Szenarien frühzeitig simulieren
- Auswirkungen von neuen Features schon vor dem Release testen

Notwendige LPT Vorbedingungen



Testbare Software



Definition der zu testenden Funktionen

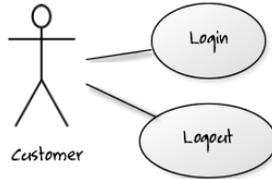
- Einzelne API Calls
- Teilbereiche der Applikation
- Ganze Anwendungsfälle
- Die „Realität“
- Und: wie stark ist die jeweilige Belastung



LPT Experte

Notwendige LPT Vorbedingungen

Was wird wie getestet



Testbare Software



LPT Experte

Passende LPT Testumgebung

- Selten gleich der fkt. Testumgebung
- Fokus auf Architektur und Konfiguration
- Realistische Testdaten (DSGVO!)
- Vorhandenes Monitoring
- Nicht unbedingt produktionsgleich

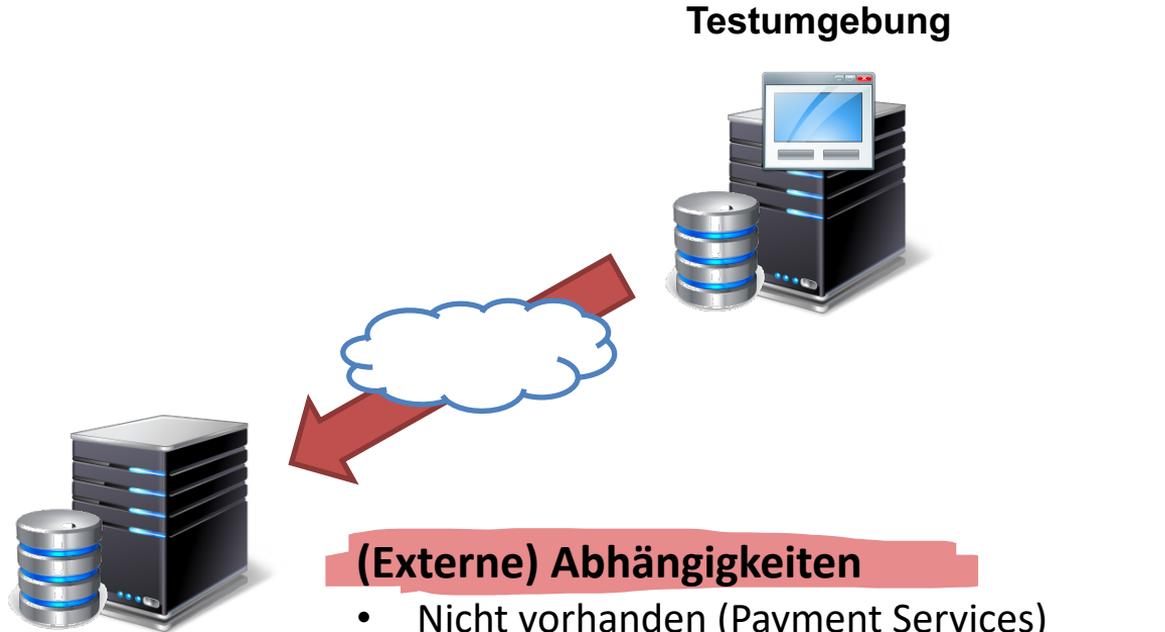
Simulation der Realität!?

Testumgebung

- Div. architektonische Unterschiede
- Andere Betriebssysteme, Patch LVL, ...
- Datenstand unterschiedlich keine Bewegungsdaten
- Untersch. Versionsstände der AUT



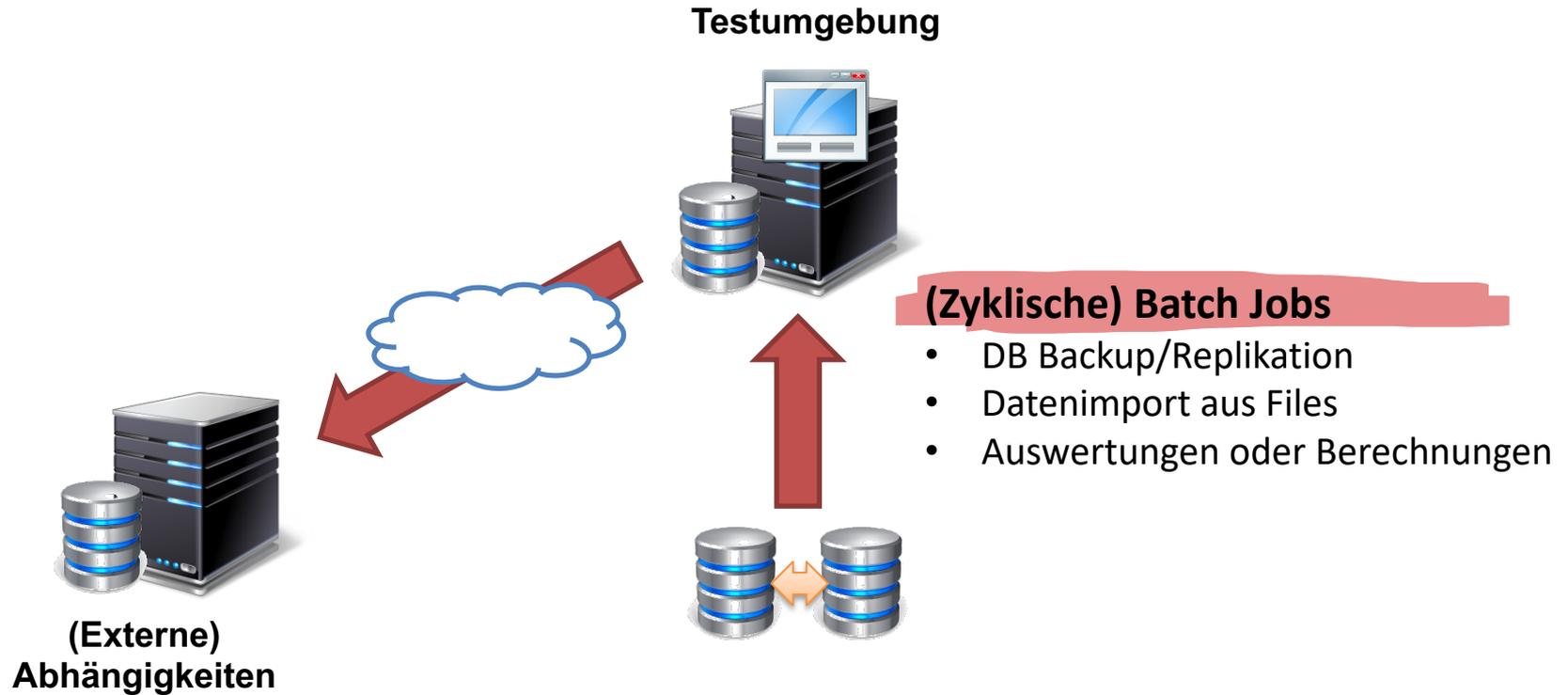
Simulation der Realität!?



(Externe) Abhängigkeiten

- Nicht vorhanden (Payment Services)
- Nicht verwendbar (nicht auf LPT ausgelegt)
- Mit zusätzlichen Kosten verbunden
- Service Virtualisierung

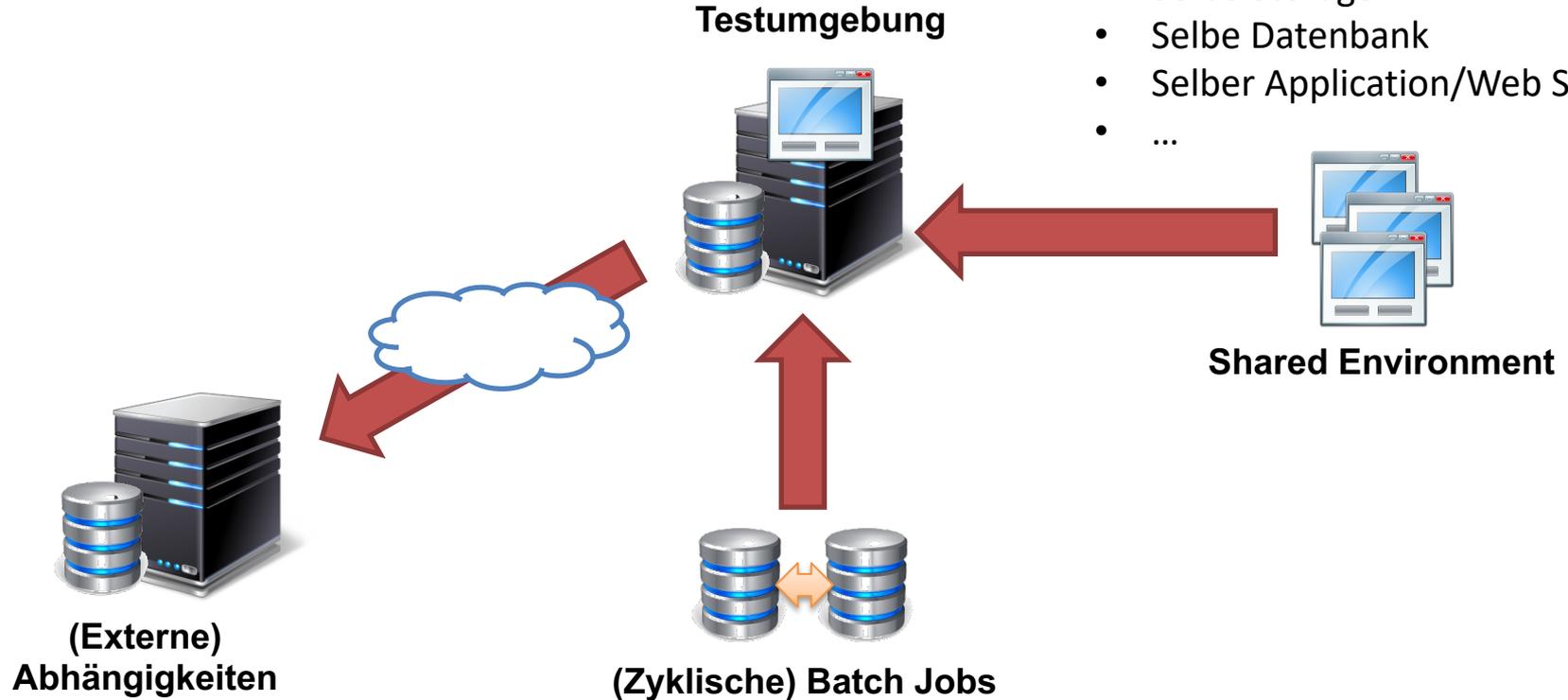
Simulation der Realität!?



Simulation der Realität!?

Shared Environment

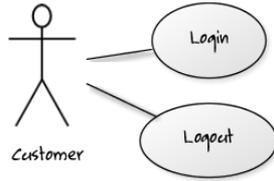
- Selber Virtualisierungs-Host
- Selbe Storage
- Selbe Datenbank
- Selber Application/Web Server
- ...



Simulation der Realität!?

Testabdeckung

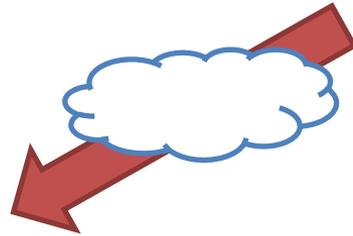
- Nur die wichtigsten Funktionalitäten/Anwendungsfälle



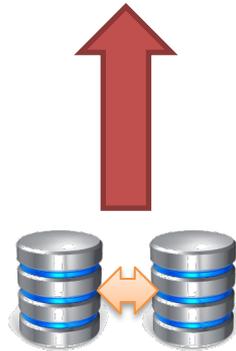
Testumgebung



Shared Environment



(Externe) Abhängigkeiten



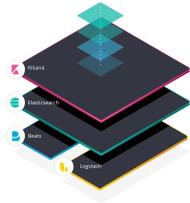
(Zyklische) Batch Jobs



5. Eine 100%ige Simulation der Realität ist nicht möglich!

- Abweichungen müssen berücksichtigt und dokumentiert werden
- Risikoabschätzung muss erfolgen
- Ergebnisse mit Bewusstsein der Abweichungen interpretieren, bewerten und kommunizieren

Monitoring: Development & Operations

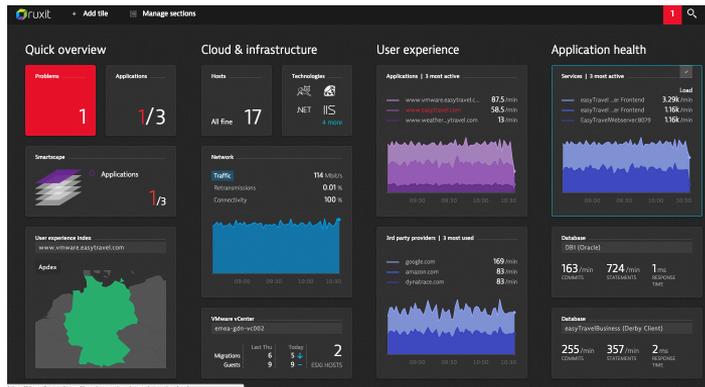
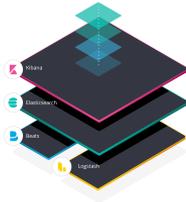


dynatrace

Nagios[®]

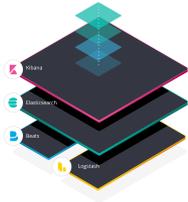
<https://grafana.com>
<https://www.elastic.io>
<https://www.dynatrace.de>
<https://www.nagios.org>

Monitoring: Development & Operations



<https://grafana.com>
<https://www.elastic.io>
<https://www.dynatrace.de>
<https://www.nagios.org>

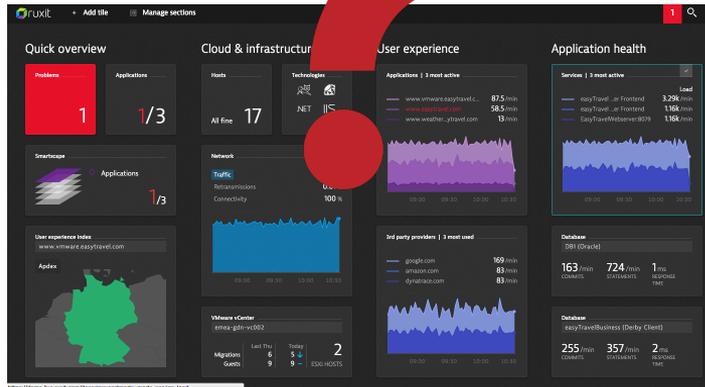
Monitoring: Development & Operations



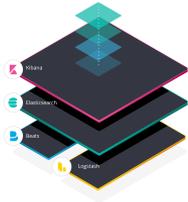
dynatrace

Nagios[®]

<https://grafana.com>
<https://www.elastic.io>
<https://www.dynatrace.de>
<https://www.nagios.org>



Monitoring: Development & Operations



dynatrace

Nagios[®]

<https://grafana.com>
<https://www.elastic.io>
<https://www.dynatrace.de>
<https://www.nagios.org>



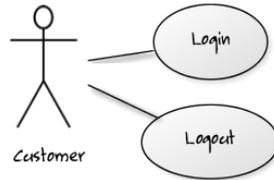


6. LPT als Vorbereitung auf den Ernstfall!

- Mit LPT die Konfiguration des Monitoring verbessern
- Die eigene Applikation kennenlernen und verstehen
- Im Ernstfall die richtigen Schlüsse ziehen und die passenden Aktionen setzen

Notwendige LPT Vorbedingungen

Was wird wie getestet



Testbare Software



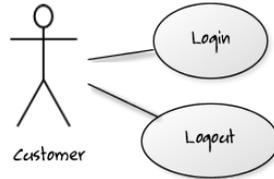
LPT Experte

Passende LPT Testumgebung

- Selten gleich der fkt. Testumgebung
- Fokus auf Architektur und Konfiguration
- Realistische Testdaten (DSGVO!)
- Vorhandenes Monitoring
- Nicht unbedingt produktionsgleich

Notwendige LPT Vorbedingungen

Was wird wie getestet



Testbare Software



Passende LPT
Testumgebung



LPT Experte



Definition der Testziele

- Welche konkrete Fragestellung?
- Wann ist der Test erfolgreich, wann nicht?
- Negativen Ergebnis – und jetzt?

Definition der Testziele: Anforderungen als Grundlage

Funktional

- Definieren das „Was“
- Vom Fachbereich/PO definiert
- Einfach umzusetzen
- Einfach zu testen
- Sofortiger Mehrwert
- Gut sichtbar für alle Benutzer

Nicht-Funktional

- Definieren das „Wie“
- Fachbereich? Architekt? Developer? Test?
- Nicht immer gezielt umsetzbar
- Hoher Testaufwand und Expertise
- Langfristiger Mehrwert
- Nur bei Problemen sichtbar

Definition der Testziele: Anforderungen als Grundlage

Funktional

- Definieren das „Was“
- Vom Fachbereich
- Einfach
- Werden (immer) spezifiziert
- Geringer Mehrwert
- Gut sichtbar für alle Benutzer

Nicht-Funktional

- Definieren das „Wie“
- Fachbereich? Anwendung? Test?
- Nicht in
- Fehlen fast immer!
- Hoher Aufwand und Expertise
- Geringer Mehrwert
- Nur bei Problemen sichtbar



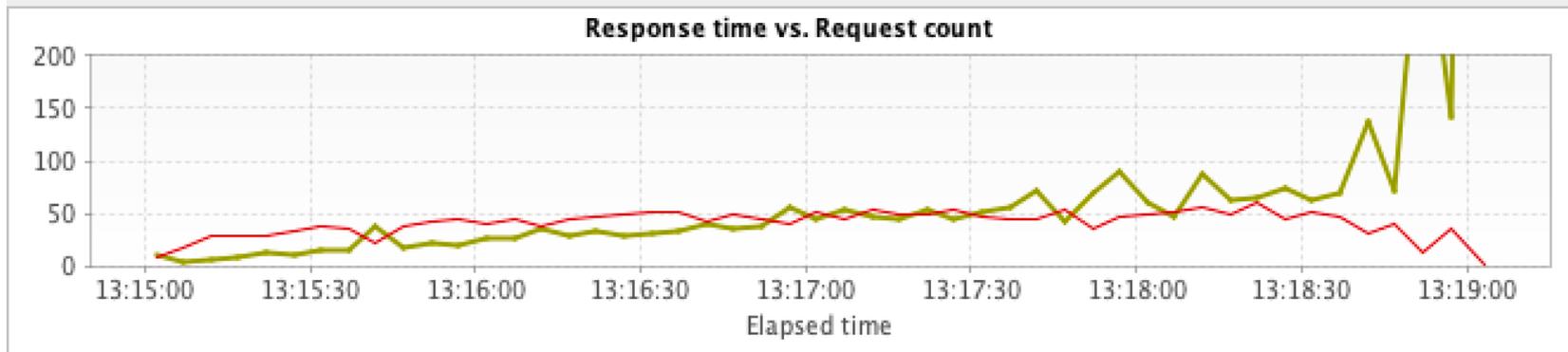
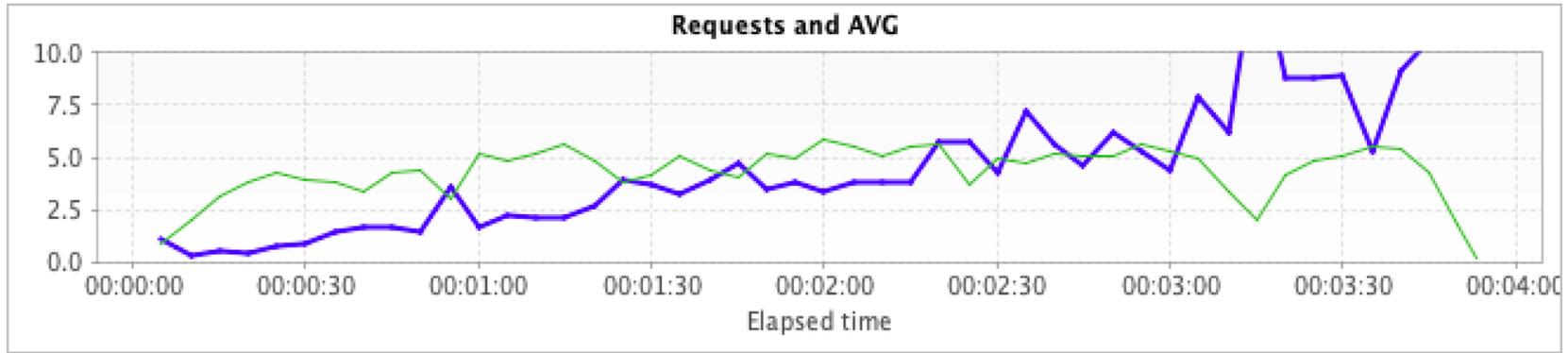
7. Achten Sie von Beginn auf die N-FKT Anforderungen!

- Ohne Nicht-Funktionale Anforderungen keine LPT-Ziele
- Ohne LPT-Ziele keine aussagekräftigen Lasttests
- Ohne Lasttests keine Absicherung der aktuellen Produktion und Simulation der zukünftiger Ereignisse

Ergebniskommunikation: was ist zu beachten?

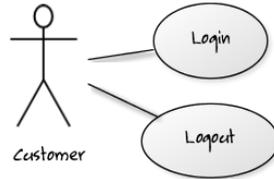
- Unklare Aussagen vermeiden
 - „x-Mal/xxx% schneller/langsamer“
 - Ohne Kontext kein Mehrwert
- Report an Stakeholder anpassen
 - Fachbereich/Development/Operations
- Nicht den Durchschnitt beachten
 - Besser: MIN, MAX, Std. Abweichung, Percentile,... beachten
- In sich konsistent
 - Graphen vergleichbar gestalten (Skala, Farben)
 - Anzahl Nachkommastellen, Sekunden vs. Millisekunden, ...

Ergebniskommunikation: was ist zu beachten?



Notwendige LPT Vorbedingungen

Was wird wie getestet



Testbare Software



Passende LPT
Testumgebung



LPT Experte

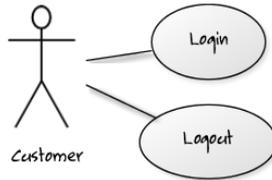


Definition der Testziele

- Welche konkrete Fragestellung?
- Wann ist der Test erfolgreich, wann nicht?
- Negativen Ergebnis – und jetzt?

Notwendige LPT Vorbedingungen

Was wird wie getestet



Testbare Software



Passende LPT
Testumgebung



LPT Experte

Passendes LPT Tool

- Zu testende Technologie (PoC)
- Lizenzkosten
- Was passt am besten zu unserer Aufgabenstellung

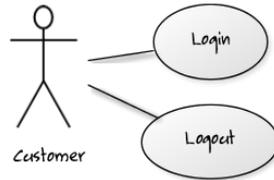


Testziel und
Erfolgskriterien



Notwendige LPT Vorbedingungen

Was wird wie getestet



Testbare Software



Passende LPT
Testumgebung



LPT Experte



Koordiniertes Vorgehen

- Wer ist involviert
- Wer wird informiert
- Was wird wann fertig
- Wer zahlt das alles

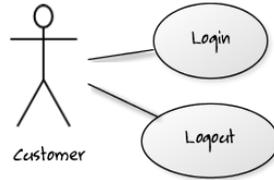
Testziel und
Erfolgskriterien



LPT Tool

Notwendige LPT Vorbedingungen

Was wird wie getestet



Testbare Software



Passende LPT
Testumgebung



LPT Experte



Aktive Planung

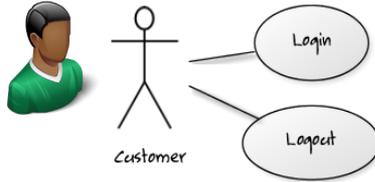
Testziel und
Erfolgskriterien



LPT Tool

Notwendige LPT Vorbedingungen

Was wird wie getestet



Testbare Software



Passende LPT
Testumgebung



LPT Experte



Aktive Planung

Testziel und
Erfolgskriterien



LPT Tool

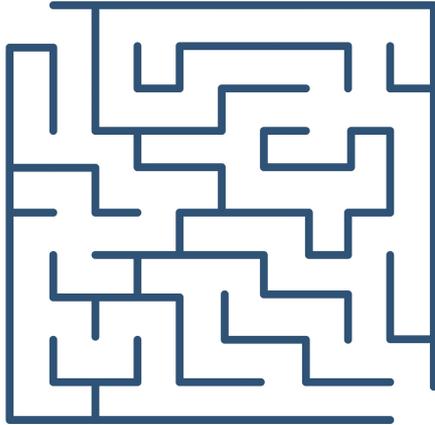


8. Last- und Performance Tests brauchen Vorbereitung!

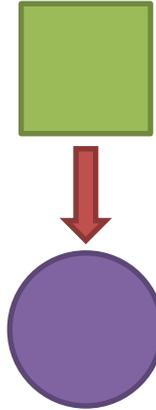
- Nicht von „heute auf morgen“ möglich
- Nicht ohne aktiver Zusammenarbeit möglich
- ABER: es ist machbar und sinnvoll!

Trigger für einen LPT

**Komplexe
Business Logic**



**Änderungen
(Config, Updates,
Frameworks)**



Go Live



CI: Regression (Performance) Testing

jenkins / customfield-editor-plugin--pipeline-master #34

Branch: customfield-editor-plugin--pipeline-master
Commit: 6647fae
No changes

22 minutes
5 days ago

Pipeline Changes Tests Artifacts

Checkout Build and Unit-Test Startup Test Environment REST-API Tests End-to-End Tests Shutdown Test Environment Cleanup

docker atlas-package
npm-install

Steps - Checkout

General SCM

CI: Regression (Performance) Testing

The screenshot displays a Jenkins pipeline execution page for 'jenkins / customfield-editor-plugin--pipeline-master #34'. The interface is primarily green, indicating a successful run. At the top, a green bar contains a checkmark icon, the pipeline name, branch 'customfield-editor-plugin--pipeline-master', commit '6647fae', and 'No changes'. On the right, it shows a duration of '22 minutes' and '5 days ago'. Below this bar are tabs for 'Pipeline', 'Changes', 'Tests', and 'Artifacts'. The main area shows a pipeline graph with steps: 'Checkout', 'Build and Unit-Test', 'Startup Test Environment', 'REST-API Tests', 'End-to-End Tests', 'Shutdown Test Environment', and 'Cleanup'. Each step has a green checkmark. A light green box highlights the 'REST-API Tests' and 'End-to-End Tests' steps. A sub-process for 'Build and Unit-Test' is shown with 'docker atlas-package' and 'npm-install' steps. At the bottom, the 'Steps - Checkout' section is expanded to show 'General SCM'.

jenkins / customfield-editor-plugin--pipeline-master #34

Branch: customfield-editor-plugin--pipeline-master
Commit: 6647fae
No changes

22 minutes
5 days ago

Pipeline Changes Tests Artifacts

Checkout Build and Unit-Test Startup Test Environment REST-API Tests End-to-End Tests Shutdown Test Environment Cleanup

docker atlas-package
npm-install

Steps - Checkout

General SCM

CI: Regression (Performance) Testing

The screenshot displays a Jenkins pipeline for a customfield-editor-plugin. The pipeline is currently successful, as indicated by the green checkmark and the green header bar. The pipeline consists of several stages: Checkout, Build and Unit-Test, Startup Test Environment, Performance Checks, Shutdown Test Environment, and Cleanup. The Performance Checks stage is highlighted with a green box and contains two sub-steps: REST-API Tests and End-to-End Tests. Below the pipeline diagram, there is a 'Response time' line graph showing the average response time in milliseconds over several builds. The graph shows a fluctuating trend with an overall downward trend. Below the graph is a table with performance metrics for various URIs.

jenkins / customfield-editor-plugin--pipeline-master #34

Branch: customfield-editor-plugin--pipeline-master
Commit: 6647fae
No changes

22 minutes
5 days ago

Pipeline Changes Tests Artifacts

Checkout Build and Unit-Test Startup Test Environment Performance Checks Shutdown Test Environment Cleanup

docker atlas-package
npm-install

REST-API Tests End-to-End Tests

Performance "Checks"

Response time

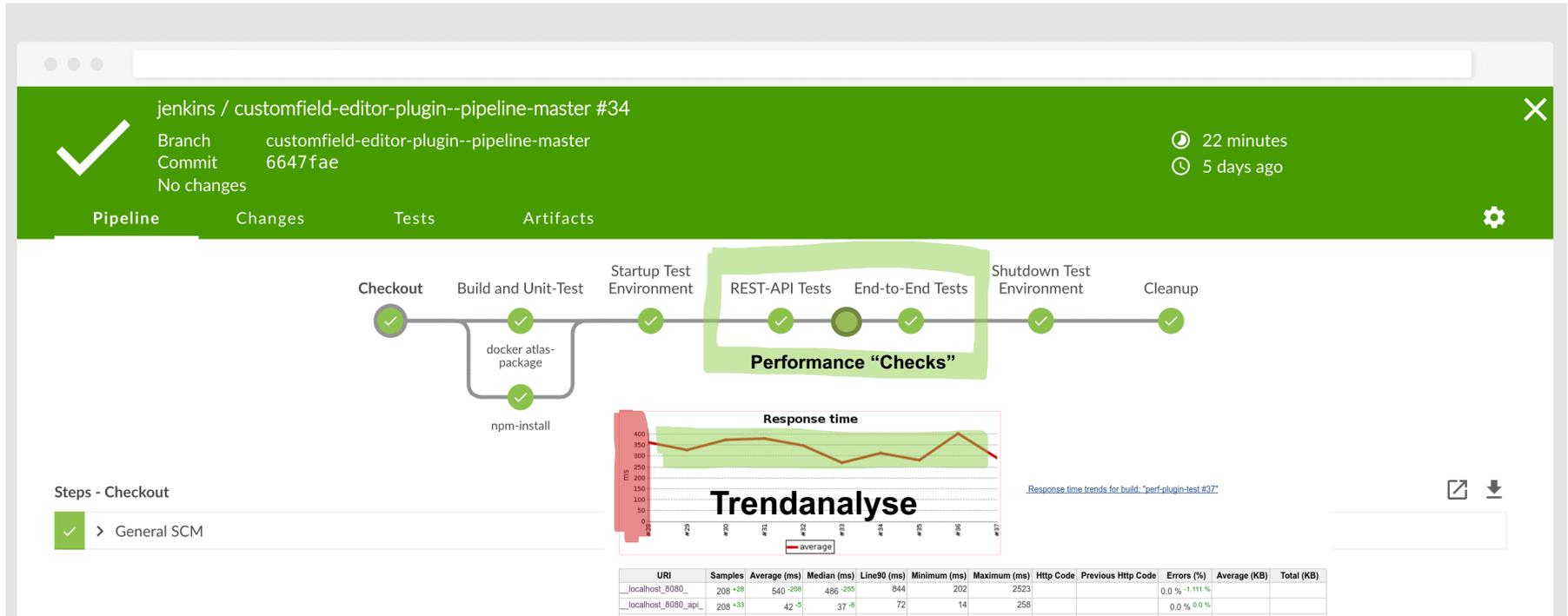
Response time trends for build: "perf-plugin-test #37"

Steps - Checkout

General SCM

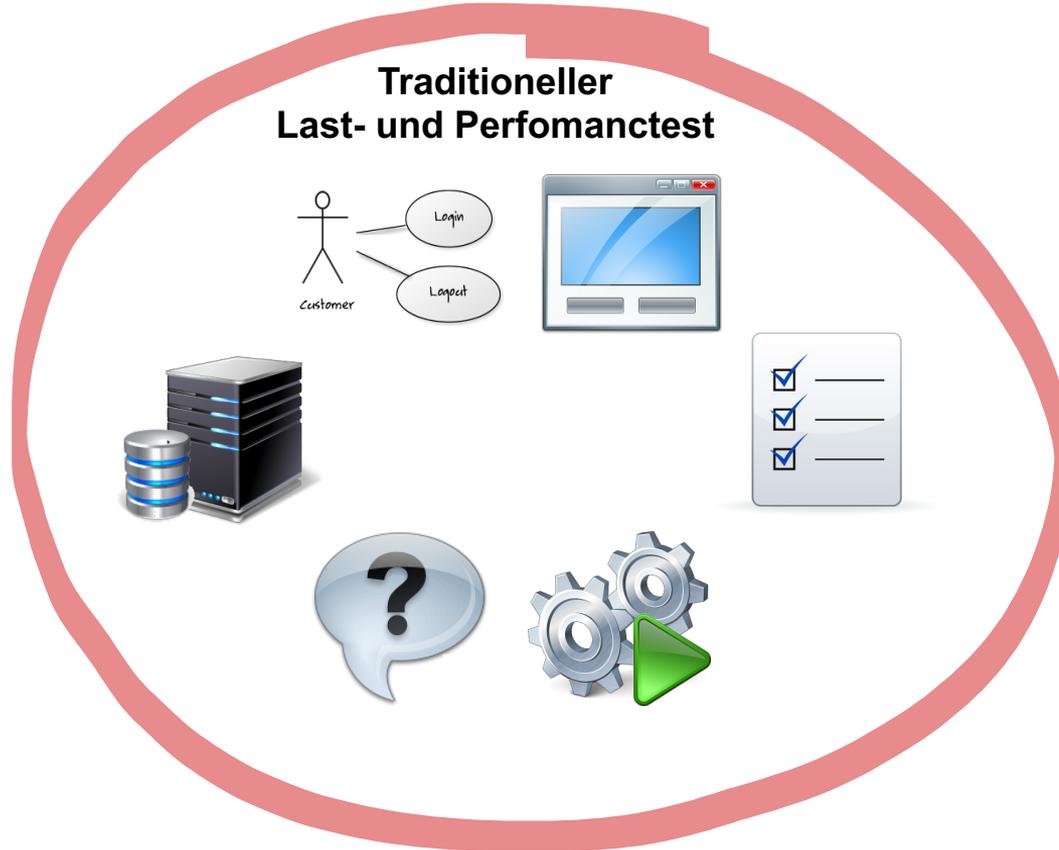
URI	Samples	Average (ms)	Median (ms)	Line90 (ms)	Minimum (ms)	Maximum (ms)	Http Code	Previous Http Code	Errors (%)	Average (KB)	Total (KB)
_localhost_8080_	208 +28	540 +208	486 +255	844	202	2523			0.0 % -1.11 %		
_localhost_8080_api_	208 +33	42 -5	37 -8	72	14	258			0.0 % 0.0 %		
All URIs	416 +61	291 -111	216 -119	763	14	2523			0.0 % -0.963 %	479903.22	1.99639739E8

CI: Regression (Performance) Testing



Performance Engineering: LPT on steroids

Traditioneller Last- und Perfomanctest



Performance Engineering: LPT on steroids

Initiale Vorbereitungen



Performance Engineering: LPT on steroids

- Testdurchführungen
 - Produktionsnahe
 - Unterstützung von Development und Operations
 - Identifikation von Bottlenecks & Performance Optimierung
 - Ergebnisdokumentation und -kommunikation

laufendes
Performance Engineering!

Initiale Vorbereitungen



Performance Engineering: LPT on steroids

- Testdurchführungen
 - Produktionsnahe
 - Unterstützung von Development und Operations
 - Identifikation von Bottlenecks & Performance Optimierung
 - Ergebnisdokumentation und -kommunikation
- Aktives Monitoring der Produktion
 - Anpassung der produktionsnahen Tests
 - Auswertungen für Operations
 - Trendanalysen
 - Analyse von Hochlastsituationen

laufendes
Performance Engineering!

Initiale Vorbereitungen



Performance Engineering: LPT on steroids

- Testdurchführungen
 - Produktionsnahe
 - Unterstützung von Development und Operations
 - Identifikation von Bottlenecks & Performance Optimierung
 - Ergebnisdokumentation und -kommunikation
- Aktives Monitoring der Produktion
 - Anpassung der produktionsnahen Tests
 - Auswertungen für Operations
 - Trendanalysen
 - Analyse von Hochlastsituationen
- Fachliche Fragestellungen erarbeiten
 - Welche neuen Funktionen sind geplant und welche Auswirkungen haben diese auf die Systemlast
 - Abstimmung und Simulation von Marketing-Aktivitäten

laufendes
Performance Engineering!

Initiale Vorbereitungen



Performance Engineering: LPT on steroids

- Testdurchführungen
 - Produktionsnahe
 - Unterstützung von Development und Operations
 - Identifikation von Bottlenecks & Performance Optimierung
 - Ergebnisdokumentation und -kommunikation
- Aktives Monitoring der Produktion
 - Anpassung der produktionsnahen Tests
 - Auswertungen für Operations
 - Trendanalysen
 - Analyse von Hochlastsituationen
- Fachliche Fragestellungen erarbeiten
 - Welche neuen Funktionen sind geplant und welche Auswirkungen haben diese auf die Systemlast
 - Abstimmung und Simulation von Marketing-Aktivitäten

laufendes
Performance Engineering!

Initiale Vorbereitungen



Ende?
EOL!

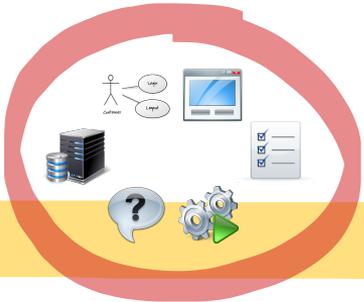
Performance Engineering: LPT on steroids

- Testdurchführungen
 - Produktionsnahe
 - Unterstützung von Development und Operations

laufendes Performance Engineering

TEAM APPROACH!

Initiale Vorbereitungen



- Trendanalysen
- Analyse von Hochlastsituationen
- Fachliche Fragestellungen erarbeiten
 - Welche neuen Funktionen sind geplant und welche Auswirkungen haben diese auf die Systemlast
 - Abstimmung und Simulation von Marketing-Aktivitäten

Ende?
EOL!



9. PE ist gelebtes Qualitätsbewusstsein!

- Kreativ, anspruchsvoll, verantwortungsvoll, interessant
- Drehscheibe verschiedener Stakeholder
- Wichtiger Bestandteil einer Teststrategie
- Team approach

Performance Engineering: What's Next?

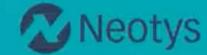
LPT & AI



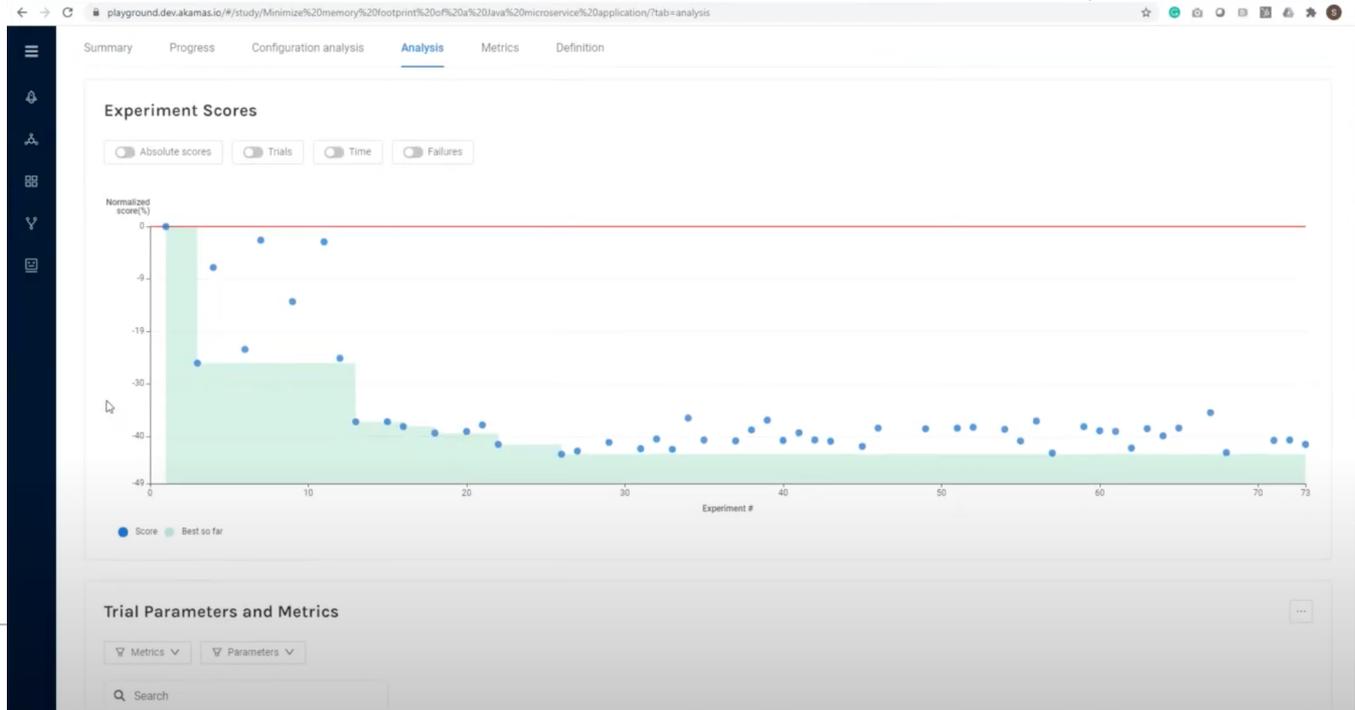
 NeoLoad AKAMAS

Performance Engineering: What's Next?

Automated Load Testing and Tuning Process



Neotys © . All Rights Reserved.



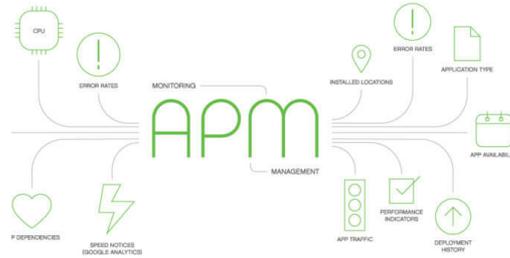
Performance Engineering: What's Next?

LPT & AI



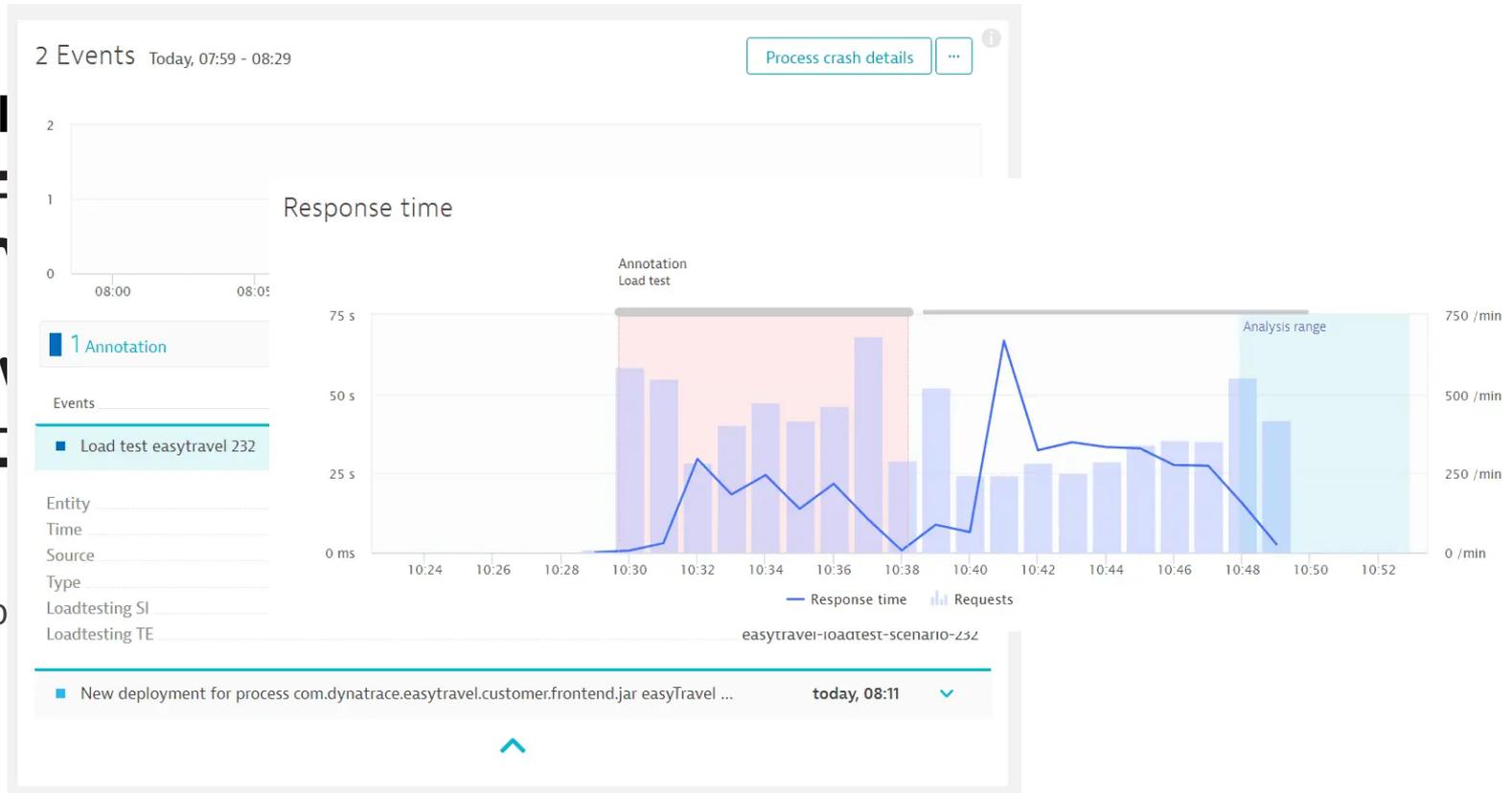
 NeoLoad 

LPT & APM



 NeoLoad 
dynatrace

Performance Engineering: What's Next?



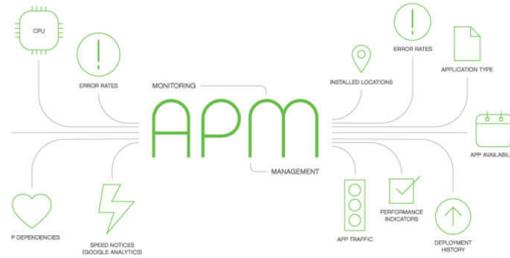
Performance Engineering: What's Next?

LPT & AI



 NeoLoad 

LPT & APM



 NeoLoad  dynatrace

Chaos Engineering



 Gremlin

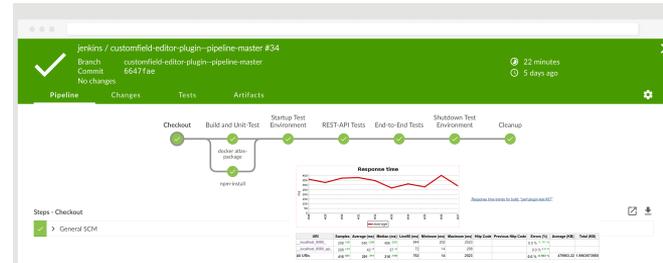
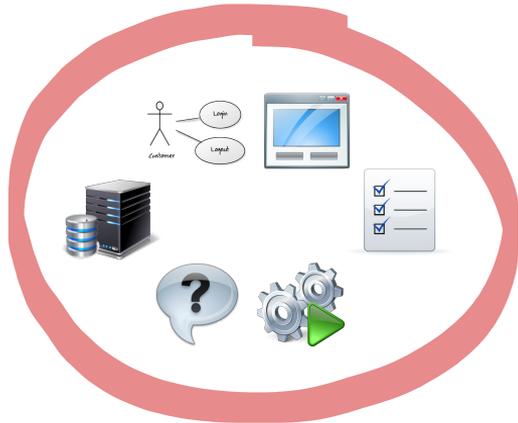


10. Blicken Sie über den Tellerrand!

- „Shift Left“ und „Shift Right“ als weitere Trigger für PE
- APM und AI in Kombination mit LPT
- Chaos Engineering als ultimativer Test der Stabilität

Performance Engineering: „Was wäre, wenn ... ?“

EINFACH MACHEN!





Ten more **things**

Performance Engineering: „Was wäre, wenn ... ?“

Klemens Loschy