

„10 things I wished they'd told me!“

aktuell. innovativ. praxisbezogen.



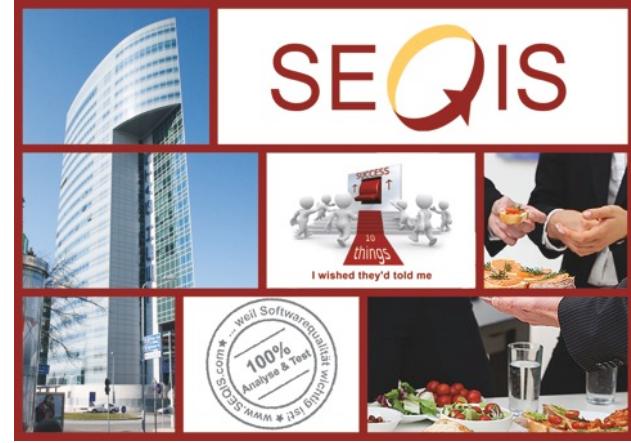
SEQIS „10 things“ – Programm 2016

17.03.2016 Kritische Erfolgsfaktoren und agile Projekte

16.06.2016 Last- und Performancetest: Jetzt!

15.09.2016 Das Internet der Dinge (IoT): Der nächste QA Paradigmenwechsel, garantiert!

17.11.2016 Collaborative Documentation: Mit just enough zu mehr Nachhaltigkeit



Last- und Performancetest: Jetzt!

Martin Wildbacher
Principal Test Consultant



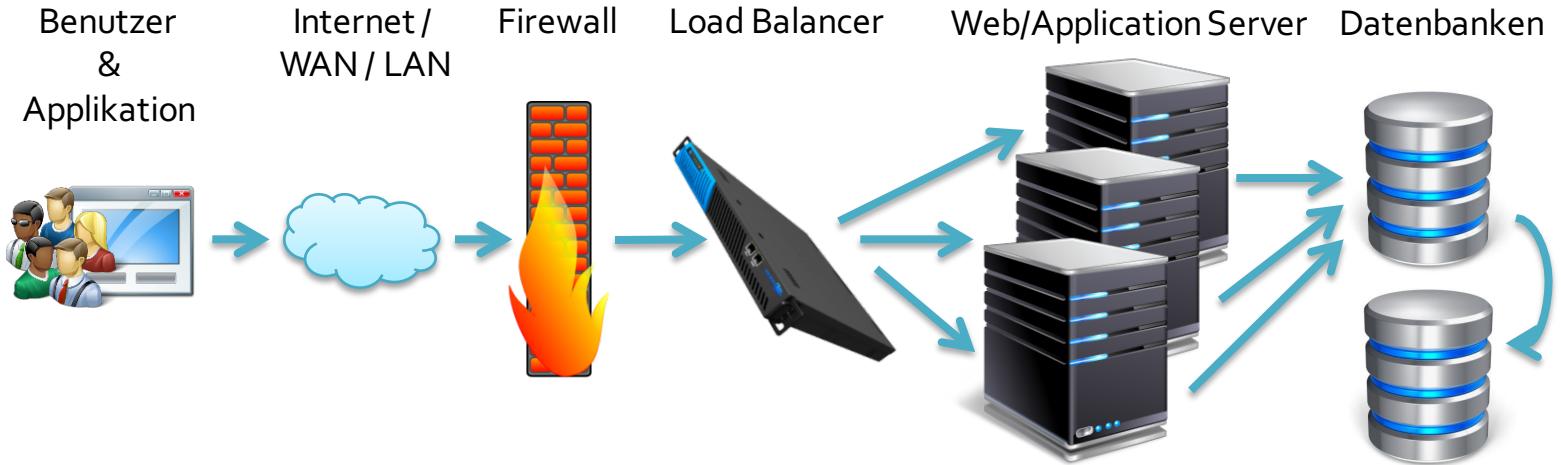
Wieso Last- und Performancetest?

Berühmte Beispiele für Server-Überlastungen:

- Spritpreis Datenbank
- Diabolo
- Apple
- Bank Austria

Last- und Performancetest

Setup und Prozess

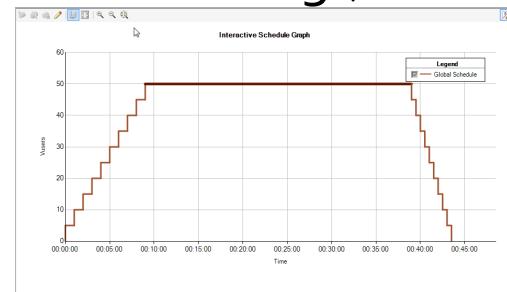


© SEQIS Software Testing GmbH

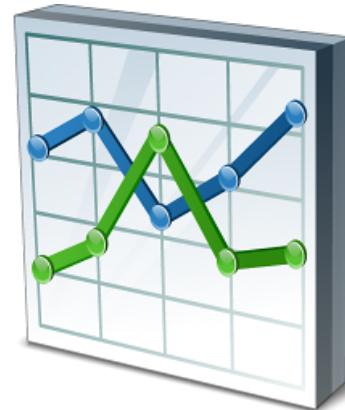
Design & scripting



Szenario Konfiguration, Durchführung & Monitoring (Controller)

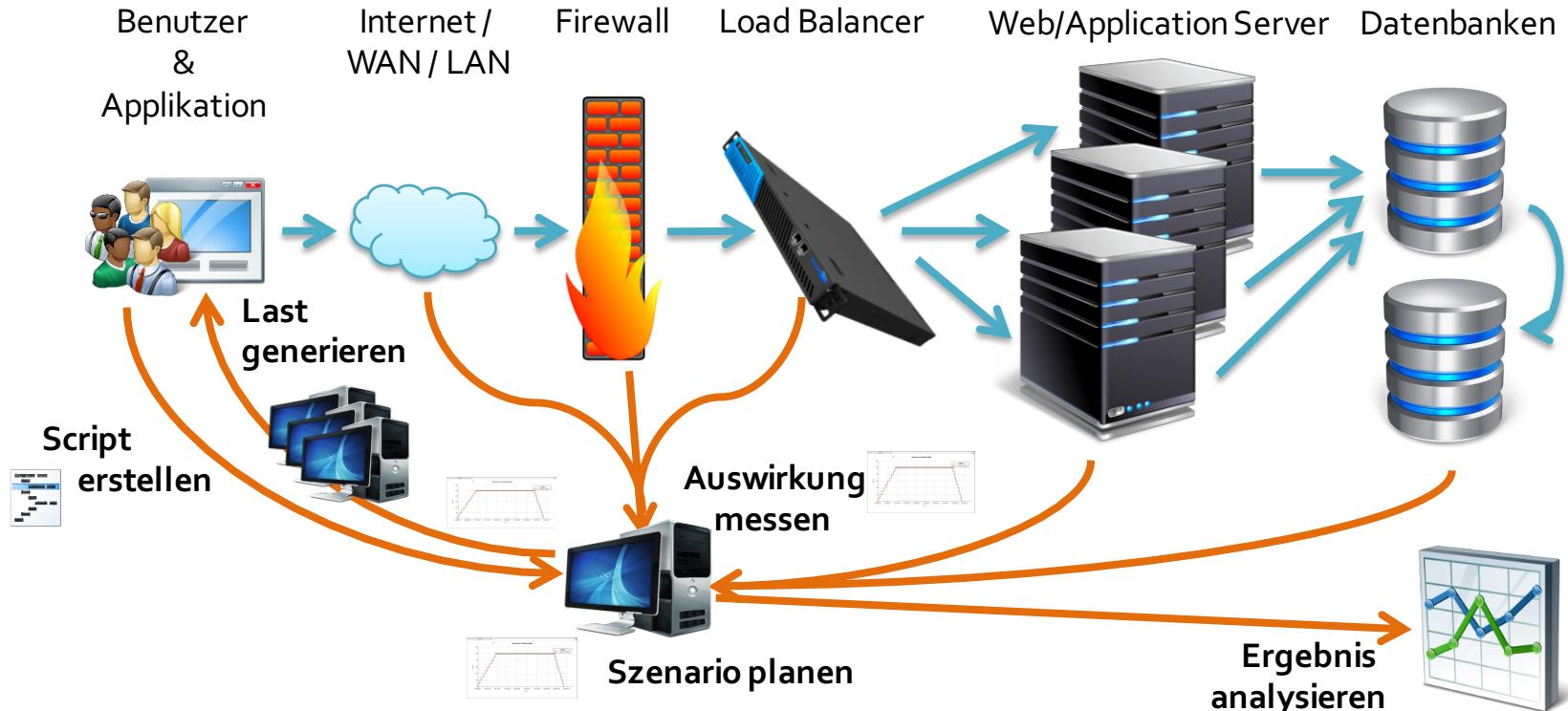


Resultatanalyse



Last- und Performancetest

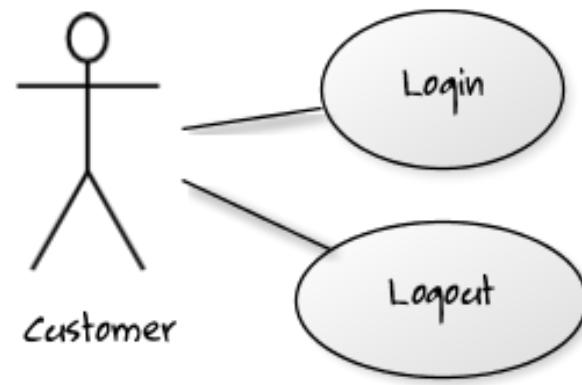
Setup und Prozess





**1. Nehmen Sie auch bei einem LPT
den Datenschutz nicht auf die
leichte Schulter**

Nur die wichtigsten Anwendungsfälle (inkl. Häufigkeit & Think Times)



2. Verproben Sie die Antworten bereits zu Beginn

- Definieren Sie konkrete Ziele
- Definieren Sie Kennzahlen

Viele verschiedene Personen involviert



3. Binden Sie rechtzeitig Experten mit ein

- Performance ist ein komplexes Gebiet
- Späte Qualitätssicherung ist teuer
- Nutzen Sie Best Practices und die Erfahrung von Profis



4. Machen Sie einen PoC

- ... das bringt Sicherheit und spart Zeit

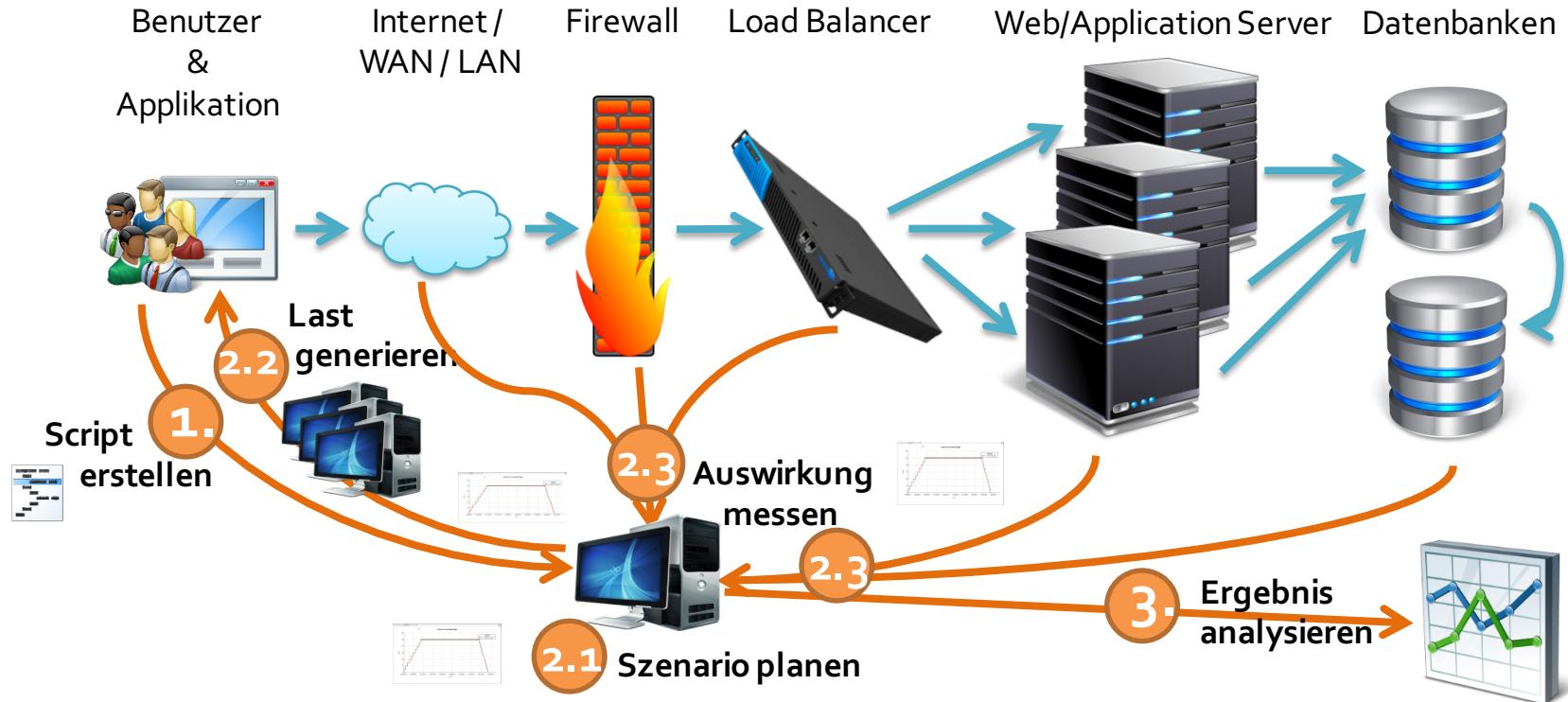
5. Achtung bei 3rd Party-Services

- zB Facebook
- Google Analytics
- Stichwort: (D)DoS-Angriff

6. Planen Sie mehrere Iterationen

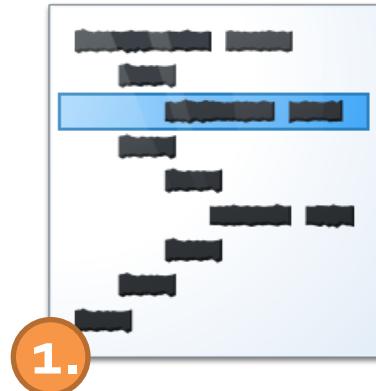
- Pro Testiteration wird üblicherweise immer nur ein Problem gefunden
- Pro Iteration immer nur eine Veränderung machen
- ... das kostet Zeit

Anforderungen an Tools (Simulation, Monitoring & Analyse)

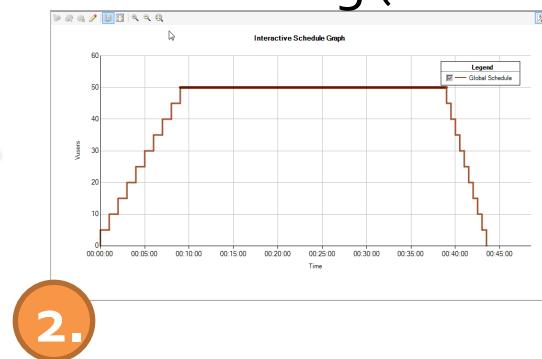


Basisanforderungen

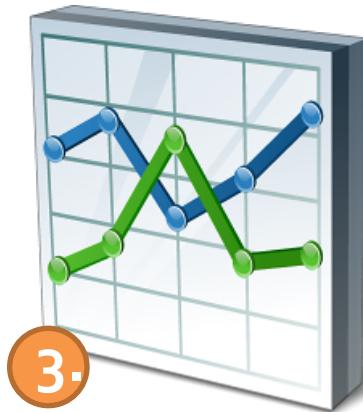
Design & scripting



Szenario Konfiguration,
Durchführung &
Monitoring (Controller)



Resultatanalyse



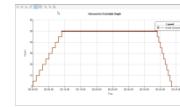
Design und Scripting

1.

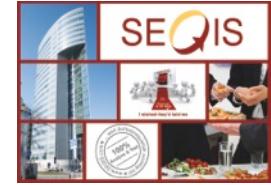
- Script Editor
 - Scriptsprache
 - IDE
 - Recording
 - Debugging
 - Error Handling
 - Dynamische Parameter
 - Response Verifikation und Extraktion
 - Transaktionen
 - Init/Action/End
- Protokolle
 - Web (HTML/HTTP)
 - SOAP
 - Citrix
 - AJAX
 - RDP
 - SAP
 - Oracle
 - Com/Dcom
 - AFP
 - TCP/IP
 - ...



Szenario Konfiguration, Durchführung & Monitoring



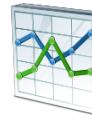
2.



- Controller
 - Flexibilität der Szenarien
 - Skalierbare und flexible Last
 - Steuerung separater Lastgeneratoren
- Monitore
 - Antwortzeiten
 - Load: Hits/s, Throughput/s
 - OS: Linux, Windows, Solaris...
 - DB: SQLServer, Oracle, MySQL, ...
 - App.Server: Tomcat, JBoss, Apache
 - **aber Echtzeit!**



Resultatanalyse



3.



- Root Cause Analyse
- Filtermöglichkeiten
- Graphische Auswertungen
- Reporterstellung
- Customizing des Reports
- Große Datenmengen
- Datenimport





Lasttestwerkzeuge

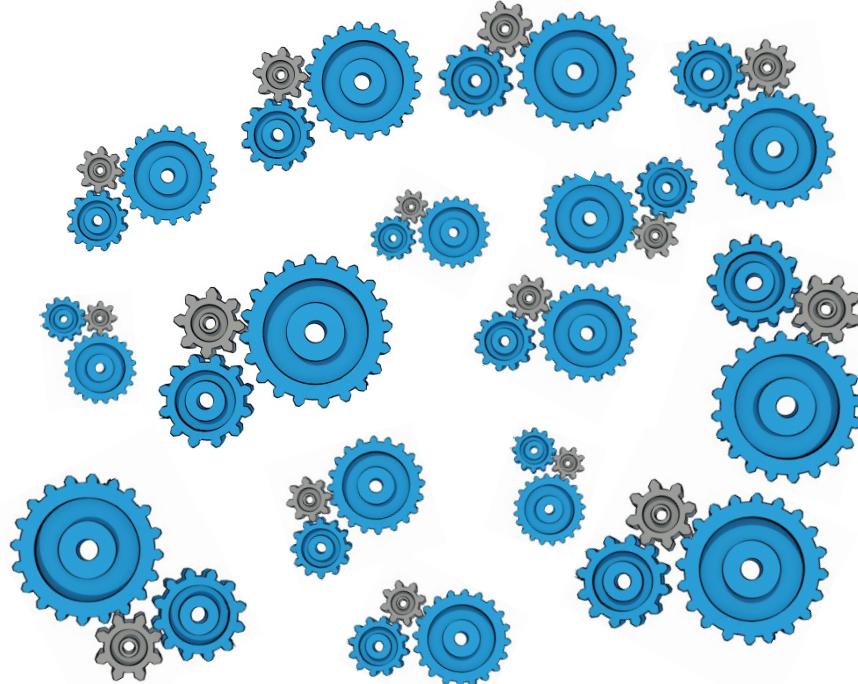
- Open Source
 - Apache Jmeter
 - The Grinder
 - OpenSTA
 - Gatling
 - ...
- Kommerziell
 - Neotys NeoLoad
 - HP LoadRunner
 - Microsoft Silkperformer
 - Radview
 - IBM Rational Performance Tester
 - Visual Studio Loadtest
 - Browsemob/Neustar (GUI)
 - ...



7. Wählen Sie ein Tool mit Echtzeitmonitoring

- ...das alle Ihre Komponenten unterstützt
- ...sonst wird eine vernünftige Analyse sehr schwer

Microservices



Microservices

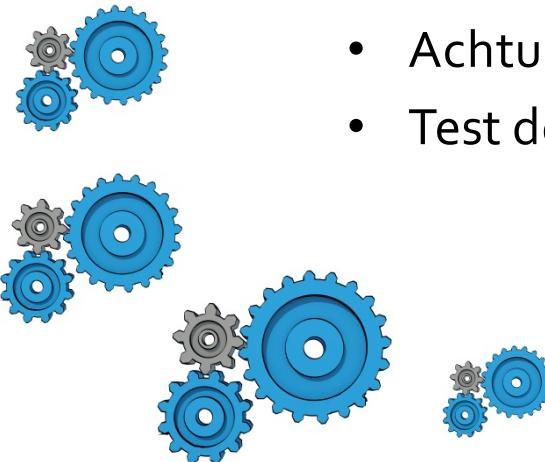
- Implizieren verteilte Systeme
- Viele Remote Procedure Calls (RPCs)
- Potentielle Probleme:
 - Netzwerk-Latenzzeit
 - Unzuverlässige Netzwerke
 - Wechselnde Last in den Applikations-Tiers
 - Lastverteilung
- Viele Prozesse auf Server





Microservices

- Performancetest von Microservices:
 - Microservices einzeln testen
 - Abhängigkeiten untereinander kennen
 - Achtung bei externen MS
 - Test der Integration in App



Internet of Things



Internet of Things

- Herausforderungen
 - Real-Time Entscheidungen notwendig
 - Anzahl der unterschiedlichen Devices
 - Unterschiedliche Einsatzbedingungen
 - Bandbreite, Packet loss, ...
 - Anzahl gleichzeitiger Verbindungen
 -



Internet of Things

- Meist TCP, UDP, Websockets
- Peer 2 Peer Protokolle (zB Z-Wave, ZigBee)
- Passendes Tool
 - Protokoll-Unterstützung
 - Bandbreiten-Beschränkung
 - Latenzsimulation
- Vorgehen wie bei „Standard“-LPT





8. Sichern Sie Ihre Investition ab

LPT Trends

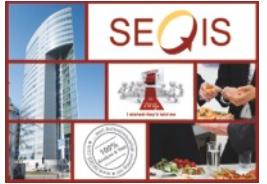
- Lasttest über das GUI
 - Vorteile:
 - Scripte meist einfacher und schneller zu erstellen
 - Protokoll-KnowHow nicht notwendig
 - Nachteile:
 - Flexibilität
 - Stabilität
 - Errorhandling schwierig
 - Weniger Virtuelle Benutzer pro Lastgenerator



LPT Trends

- Einbetten in Continuous Integration
 - Stichwort Profiling
 - „Build to Build“-Vergleich
 - Trend-Charts





9. Nutzen Sie die passenden Tools

- Gehen Sie davon aus, dass Sie für Performance Engineering zumindest zwei Tools verwenden werden.

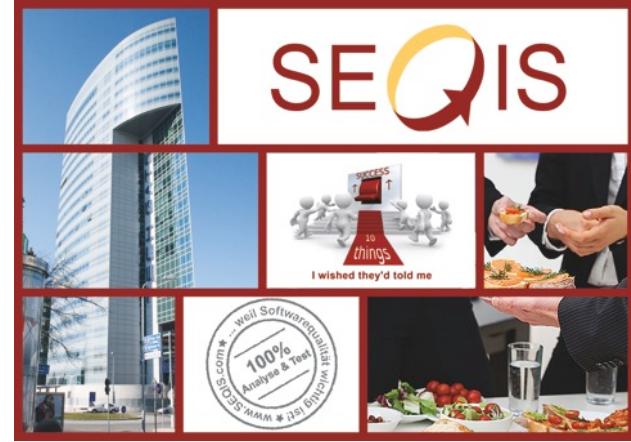


**10. Starten Sie JETZT Ihren nächsten
Last- und Performancetest!**



Zusammenfassung

1. Nehmen Sie auch bei einem LPT den Datenschutz nicht auf die leichte Schulter
2. Verproben Sie die Antworten bereits zu Beginn
3. Binden Sie rechtzeitig Experten mit ein
4. Machen Sie einen PoC
5. Achtung bei 3rd Party-Services
6. Planen Sie mehrere Iterationen
7. Wählen Sie ein Tool mit Echtzeitmonitoring
8. Sichern Sie Ihre Investition ab
9. Nutzen Sie die passenden Tools
10. Starten Sie JETZT Ihren nächsten Last- und Performancetest!



Last- und Performancetest: Jetzt!

Martin Wildbacher
Principal Test Consultant



SEQIS „10 things“ – Programm 2016

17.03.2016

Kritische Erfolgsfaktoren und agile Projekte

16.06.2016

Last- und Performancetest: Jetzt!

15.09.2016

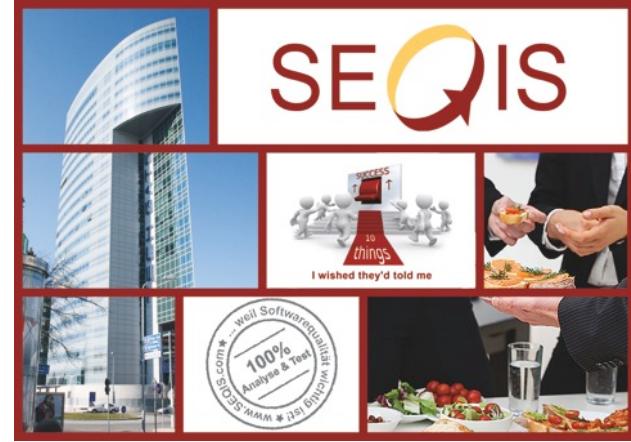
**Das Internet der Dinge (IoT): Der nächste QA
Paradigmenwechsel, garantiert!**

17.11.2016

Collaborative Documentation: Mit just enough
zu mehr Nachhaltigkeit

16.06.16

„10 things“ – Last- und Performancetest: Jetzt!



„10 things I wished they'd told me!“

aktuell. innovativ. praxisbezogen.