

Mit Performance-Modellierung Test- und Betriebskosten senken



Dr. Andreas Brunnert
CEO
RETIT



Peter Zahrer
Produktmanager
Dynatrace

Agenda

Performance-Modellierung

Anwendungsbeispiele, Nutzen und
Kundenszenarien

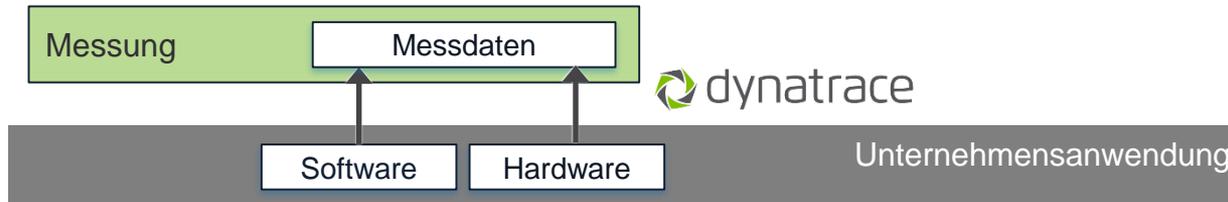
Demo



Warum Performance-Modellierung?

Moderne Applikationen benötigen die Fähigkeit

- ... Die Performance von Anwendungen in Echtzeit zu messen
- ... Performance-Metriken in Echtzeit auszuwerten
- ... Drill Down in einzelne Komponenten durchzuführen



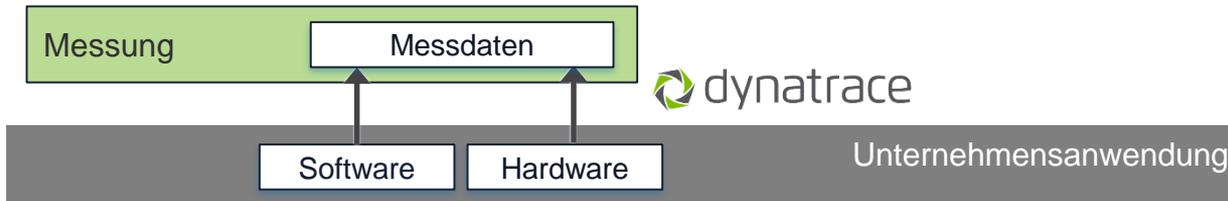
Warum Performance-Modellierung?



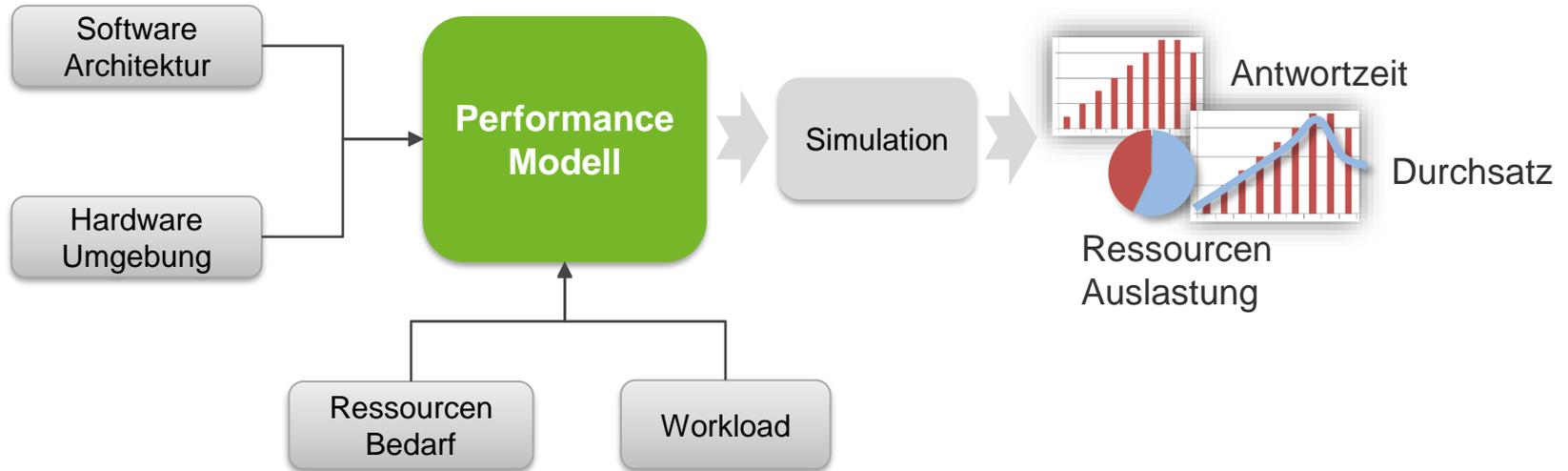
Was passiert wenn ...



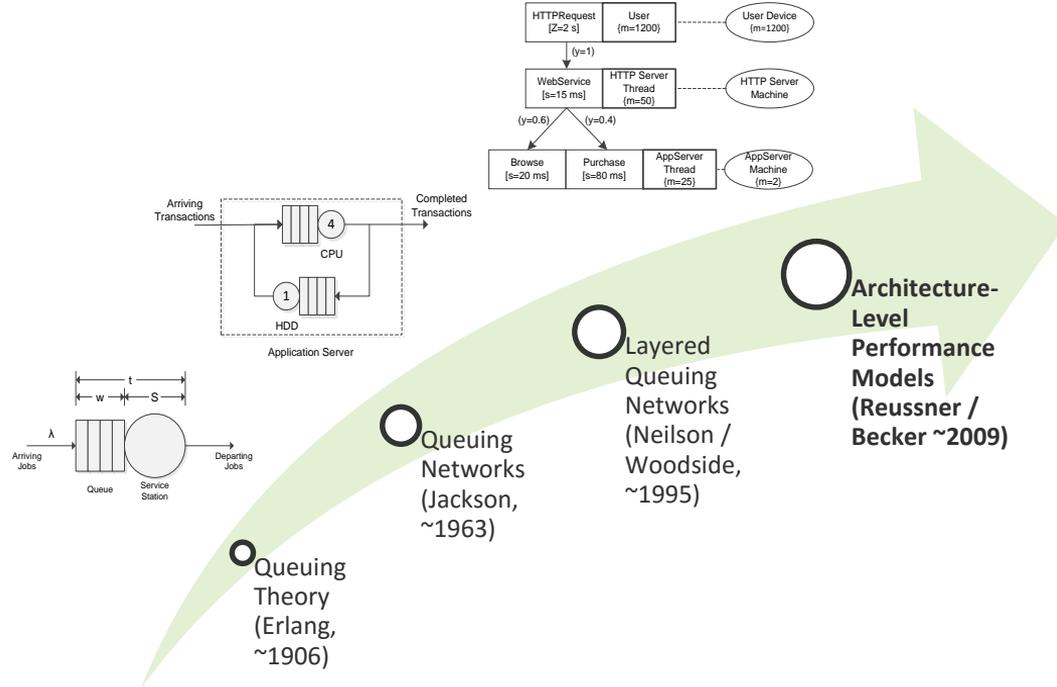
- ... man die Deployment Topology ändert
- ... man auf eine andere Hardware Umgebung migriert?
- ... sich die Last ändert?
- ... man die Anzahl der CPUs reduziert?



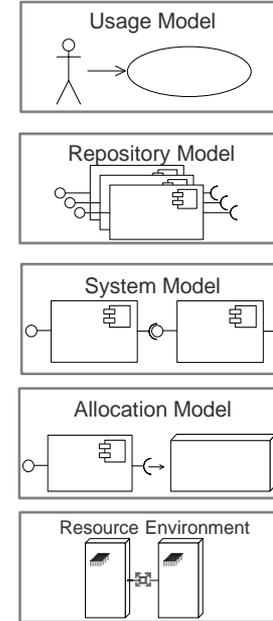
Performance-Modellierung – Was ist das?



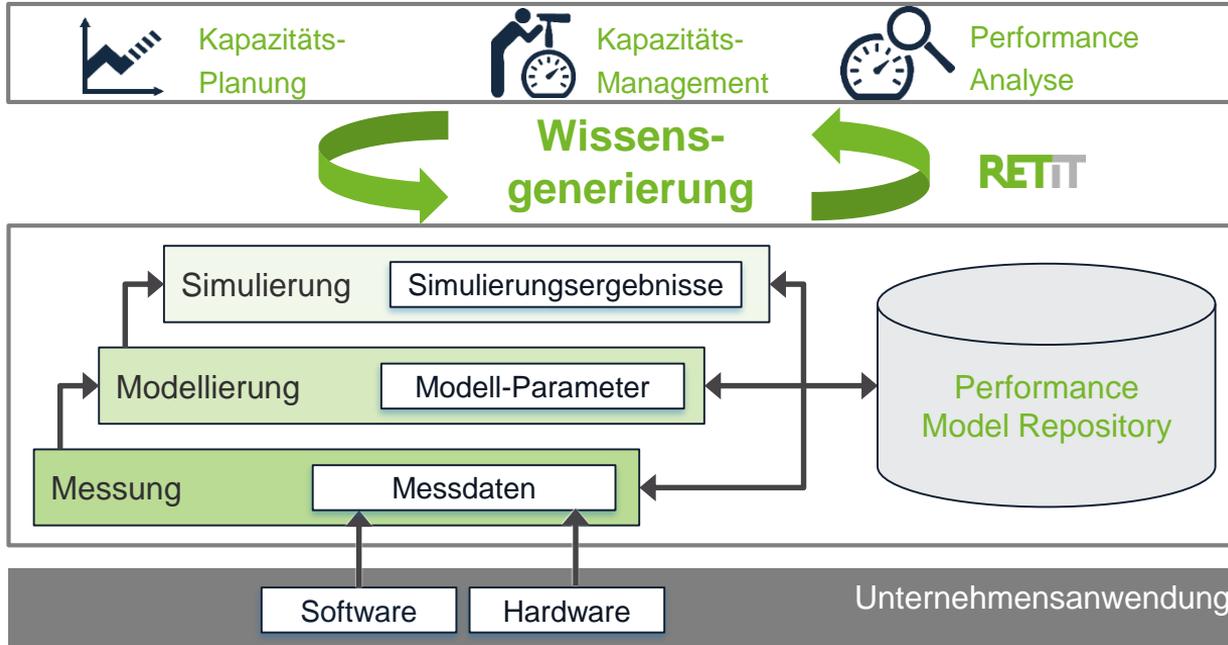
Performance-Modellierung – Ein kurzer Rückblick



Palladio Component Model (PCM)



Performance-Modellierung – Wie?



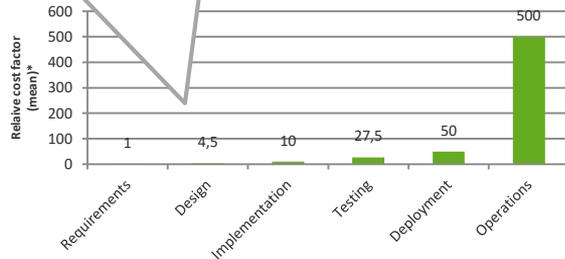
Performance-Modellierung – Wann?

Dev Ops

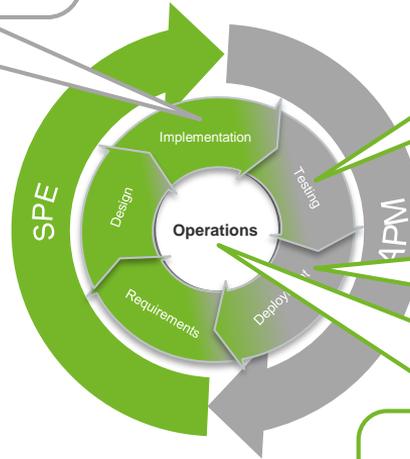
Feststellung von Performance-Änderungen in jeder Version, die in der Continuous Delivery Pipeline erzeugt wird, ohne kostenintensive Performance-Tests durchzuführen

Last- und Performance-Testergebnisse extrapolieren, Anzahl der Testläufe, Aufwand, Hardware- und Lizenzkosten reduzieren

Mit dem Wissen aus Performance-Modellen Performanceprobleme schneller erkennen und diese früher und kosteneffizienter beheben



Source: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100036670.pdf>
 *The cost factor for fixing a performance-related problem is normalized in the different phases relative to the cost of fixing a defect in the requirements phase.



Kapazitätsplanung mit Modellen der Application-Binaries vereinfachen

Richtige Kapazität für saisonal bedingte Spitzen ermitteln um Lizenz- und operationale Kosten (z.B. die Anzahl Cloud Instanzen) zu reduzieren

Dynatrace kompatible RETIT Lösungen

- **RETIT Capacity Manager (RCM)**



- Performance-Modellierungsumgebung
- Lässt sich mit jedem Dynatrace Deployment verbinden um Performance Modelle zu erstellen
- Simuliert “Was-wäre-Wenn” Szenarien:
 - Änderung der Hardware
 - Änderung des Workloads
 - Änderung in der Software-Architektur

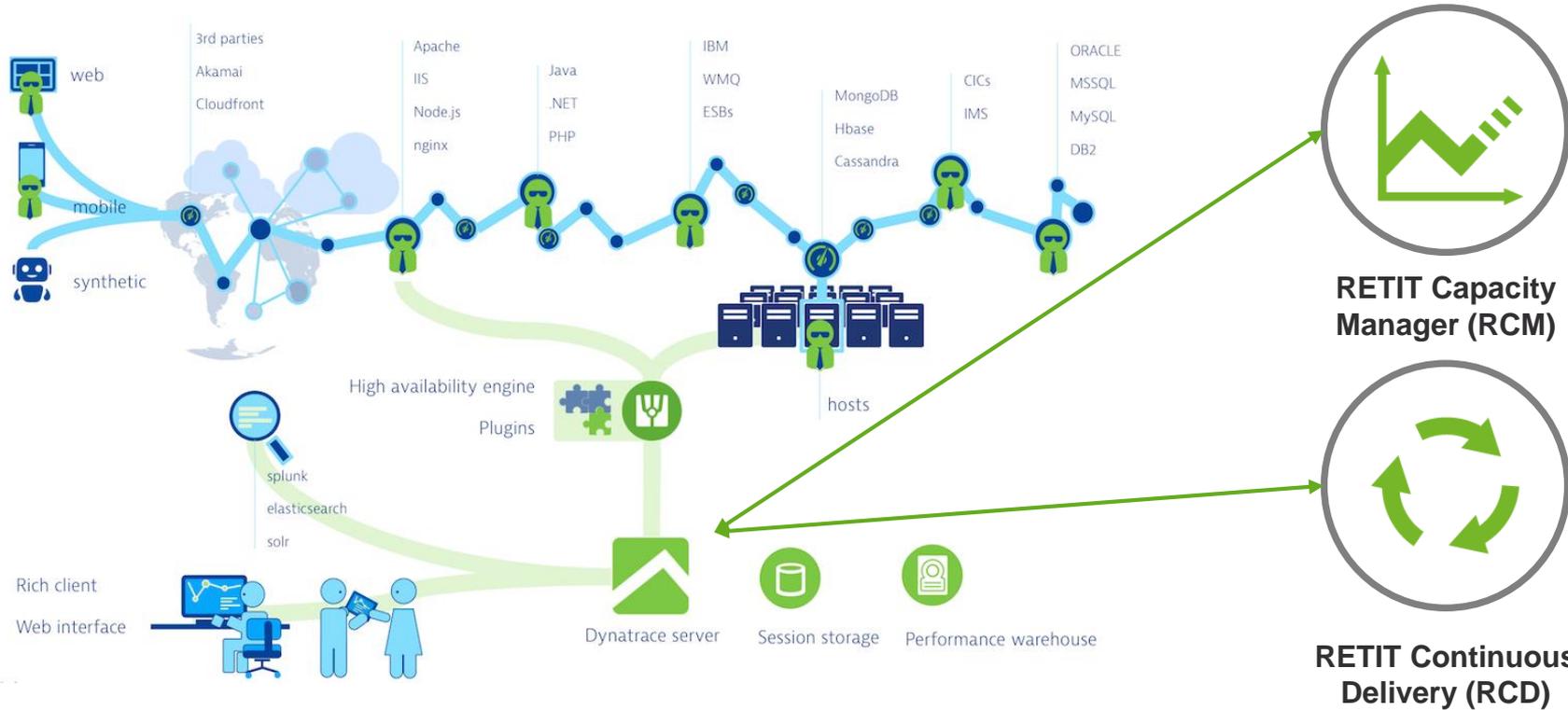
Dynatrace kompatible RETIT Lösungen

- **RETIT Continuous Delivery (RCD)**



- Plugin für Jenkins Continuous Integration (CI) Server
- Erkennt Änderungen der Performance automatisch
- Verwendet Dynatrace Daten, die in früheren Testläufen gesammelt wurden um Modelle zu erstellen
- Evaluiert die Performance für multiple Hardware-Umgebungen und Workloads

Dynatrace compatible RETIT Lösungen



Agenda

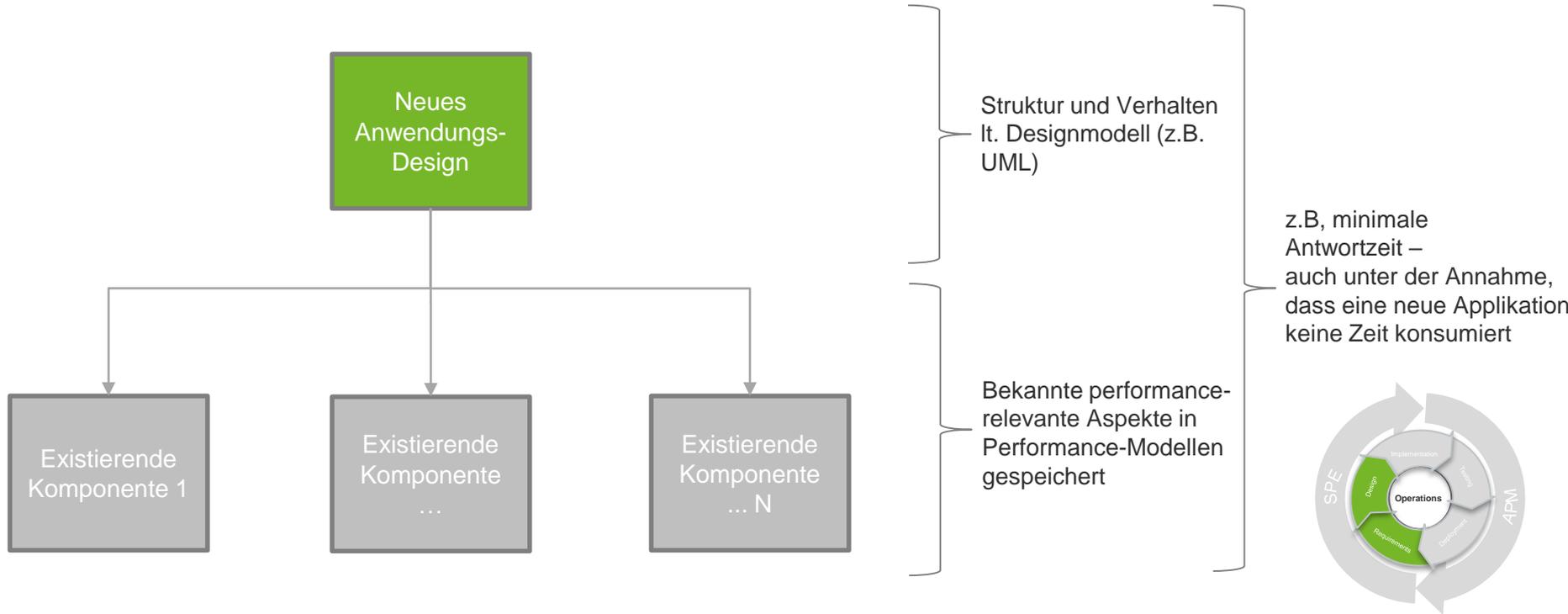
Performance-Modellierung

Anwendungsbeispiele, Nutzen und
Kundenszenarien

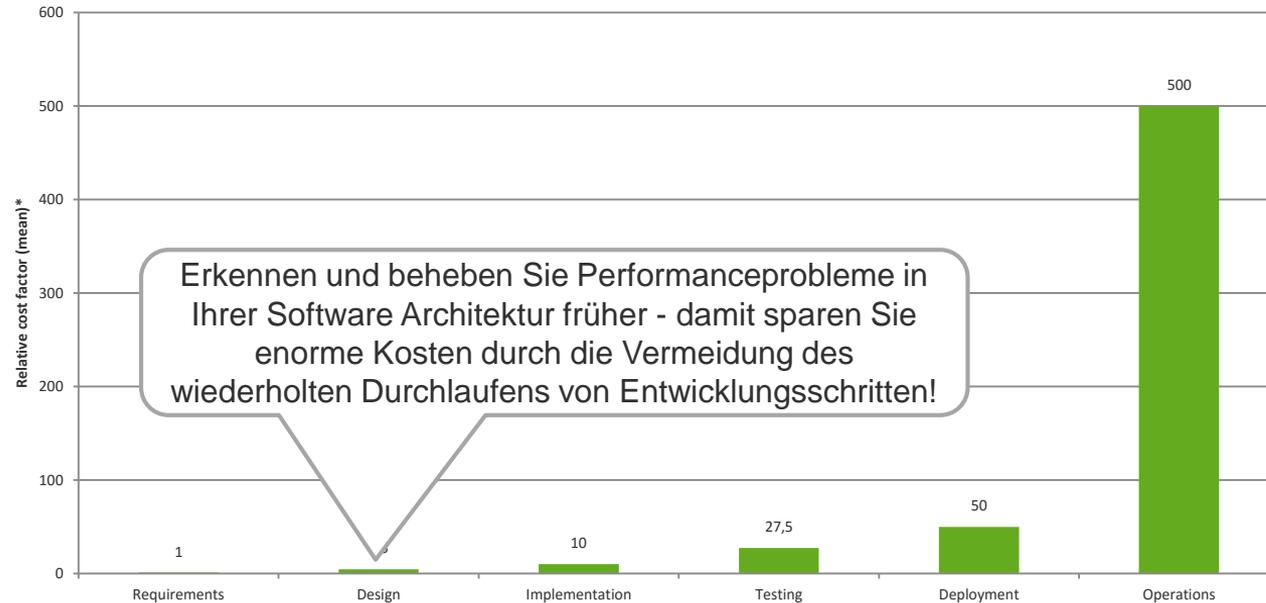
Demo



Anwendungsbeispiel – Einsatz zur Entwurfszeit



Nutzen – Einsatz zur Entwurfszeit



Source: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100036670.pdf>
 *The cost factor for fixing a performance-related problem is normalized in the different phases relative to the cost of fixing a defect in the requirements phase.



Kundenszenario – Einsatz zur Entwurfszeit

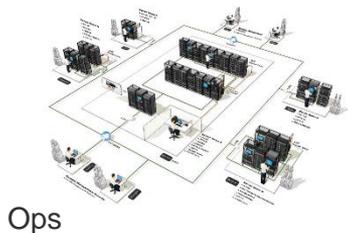
- Performance-Evaluierung in einer Service-Orientierten Architektur (SOA)



- Service-Konsumenten:**
 - Prozessorientierte User Interfaces
 - Orchestriert mit einer BPM Engine

Enterprise Service Bus (ESB)

- Service-Anbieter:**
 - Datenquellen und Anwendungsdienste



Kundenszenario – Einsatz zur Entwurfszeit

- Performance-Evaluierung in einer Service-Orientierten Architektur (SOA)



- **Service-Konsumenten:**

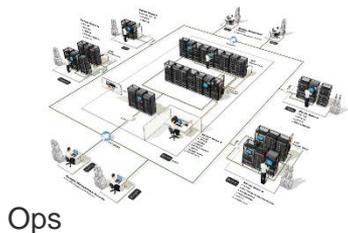
- Prozessorientierte User Interfaces
- Orchestriert mit einer BPM Engine

- Was passiert wenn wir einen neuen Business Prozess einführen oder optimieren?
- Können erforderliche KPIs des Business Prozesses eingehalten werden?
- Was kostet es, die Systemperformance zu steigern um die KPIs des Businessprozesses zu gewährleisten?



- **Service-Anbieter:**

- Datenquellen und Anwendungsdienste



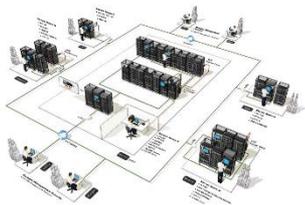
Kundenszenario – Einsatz zur Entwurfszeit

- Performance-Evaluierung in einer Service-Orientierten Architektur (SOA)



- Service-Konsumenten:**
 - Prozessorientierte User Interfaces
 - Orchestriert mit einer BPM Engine

- Können wir die gewünschte Performance erreichen und die KPIs mit den bestehenden SLAs gewährleisten?
- Welche Services müssen zunächst optimiert werden um die Ziele des Business zu erreichen?
- Müssen wir die SLAs mit mehreren Service-Anbietern anpassen oder sollen wir die SLAs mit wenigen Anbietern verschärfen?



- Service-Anbieter:**
 - Datenquellen und Anwendungsdienste



Kundenszenario – Einsatz zur Entwurfszeit

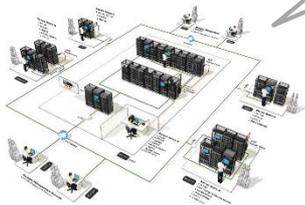
- Performance-Evaluierung in einer Service-Orientierten Architektur (SOA)



- Service-Konsumenten:**

- Prozessorientierte User Interfaces
- Orchestriert mit einer BPM Engine

- Wieviel Last wird ein neuer oder zusätzlicher Business Prozess erzeugen?
- Werden die bestehenden Systeme in der Lage sein, die Last zu bewältigen?
- Müssen wir unsere Kapazitäten erhöhen?
- Wer bezahlt die zusätzlichen Kapazitäten?



- Service-Anbieter:**

- Datenquellen und Anwendungsdienste

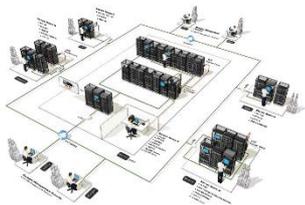


Kundenszenario – Einsatz zur Entwurfszeit

- Performance-Evaluierung in einer Service-Orientierten Architektur (SOA)



- **Service-Konsumenten:**
 - Prozessorientierte User Interfaces
 - Orchestriert mit einer BPM Engine



Ops

- **Service-Anbieter:**
 - Datenquellen und Anwendungsdienste



Kundenszenario – Einsatz zur Entwurfszeit

- Performance-Evaluierung in einer Service-Orientierten Architektur (SOA)



- Service-Konsumenten:**

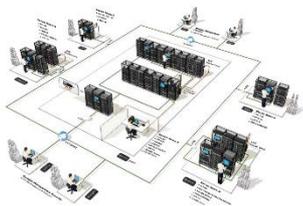
- Prozessorientierte User Interfaces
- Orchestriert mit einer BPM Engine

Die KPIs des Business Prozesses können erreicht werden. Die Kosten dafür sind ...!

- Die Systemlast wird um X% steigen!
- Es müssen X neue Server angeschafft werden.



Wir müssen mit Provider XYZ sprechen!



- Service-Anbieter:**

- Datenquellen und Anwendungsdienste

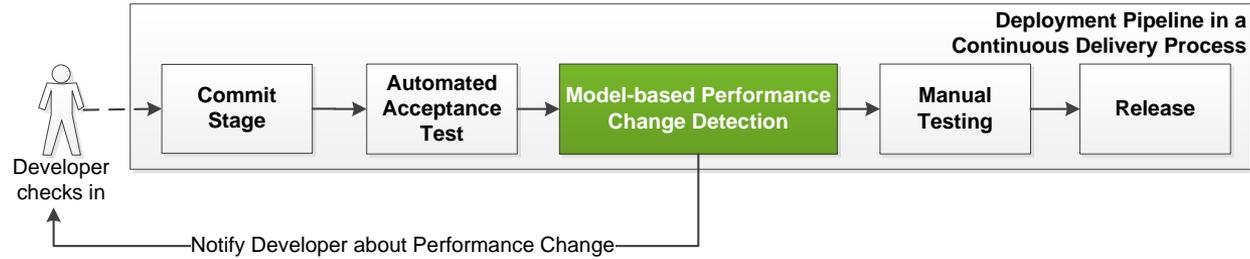


Kundenszenario – Einsatz zur Entwurfszeit

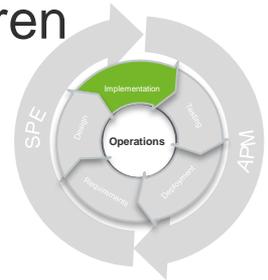
- ✓ Modellierung **steigert das Teamwork** im **gesamten Software Lifecycle** (Business, Dev and Ops)!
- ✓ Das Management erhält detaillierte Einblicke in die Business-Prozesse. **Dies erleichtert die Kommunikation mit dem IT Department.**
- ✓ Entwickler können **Antwortzeiten für neue Businessprozesse besser abschätzen.**
- ✓ Der IT-Betrieb erhält **frühzeitig Informationen** über das **Lastverhalten wenn neue Business Prozesse in Betrieb gehen.**



Anwendungsbeispiel – Einsatz im CI-System



- ✓ Auswirkungen von neuen Features oder Bug Fixes evaluieren
- ✓ Für unterschiedliche Hardwareumgebungen und Lastszenarien
- ✓ Ohne reale Test-Systeme, rein auf Basis von Modellen

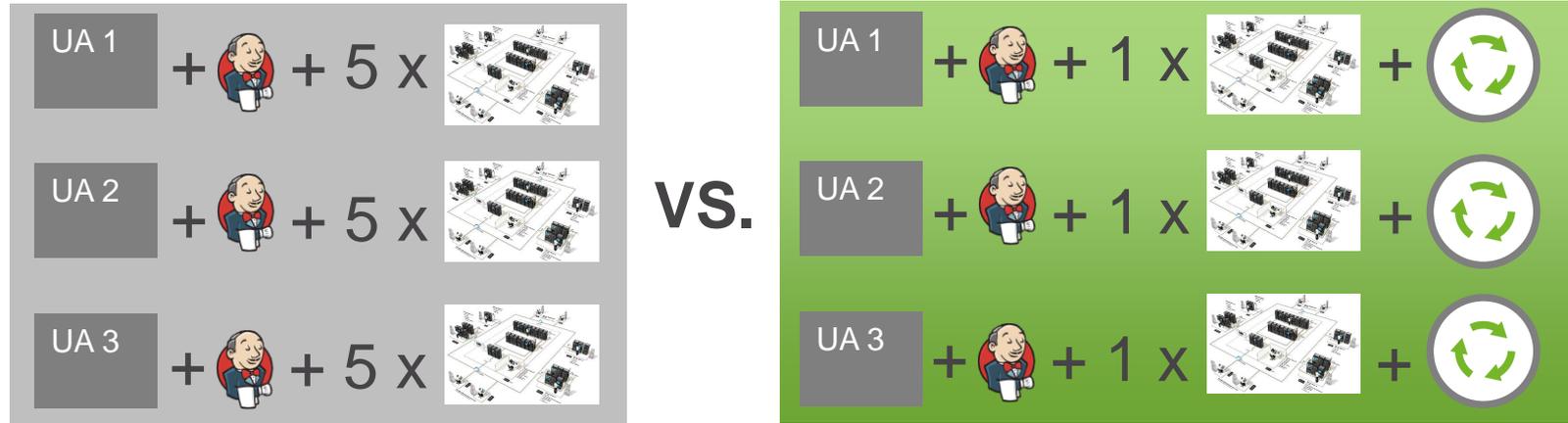


Nutzen – Einsatz im CI-System

- ✓ Sicherstellen, dass **keine Version Performanceregressionen** verursacht
- ✓ **Kostensparnis**, wenn **Performance-Probleme** früher im **Entwicklungsprozess** erkannt werden
- ✓ Besseres **Performancebewusstsein** von **Entwicklern** durch **sofortiges Feedback**
- ✓ **Kein(e) Setup/Vorbereitung** für **Last/Performance Testumgebungen** für jedes Projekt
- ✓ **Wiederverwendung** von **Akzeptanz/Regression Testing** Ergebnisse für die **Performance-Evaluierung**



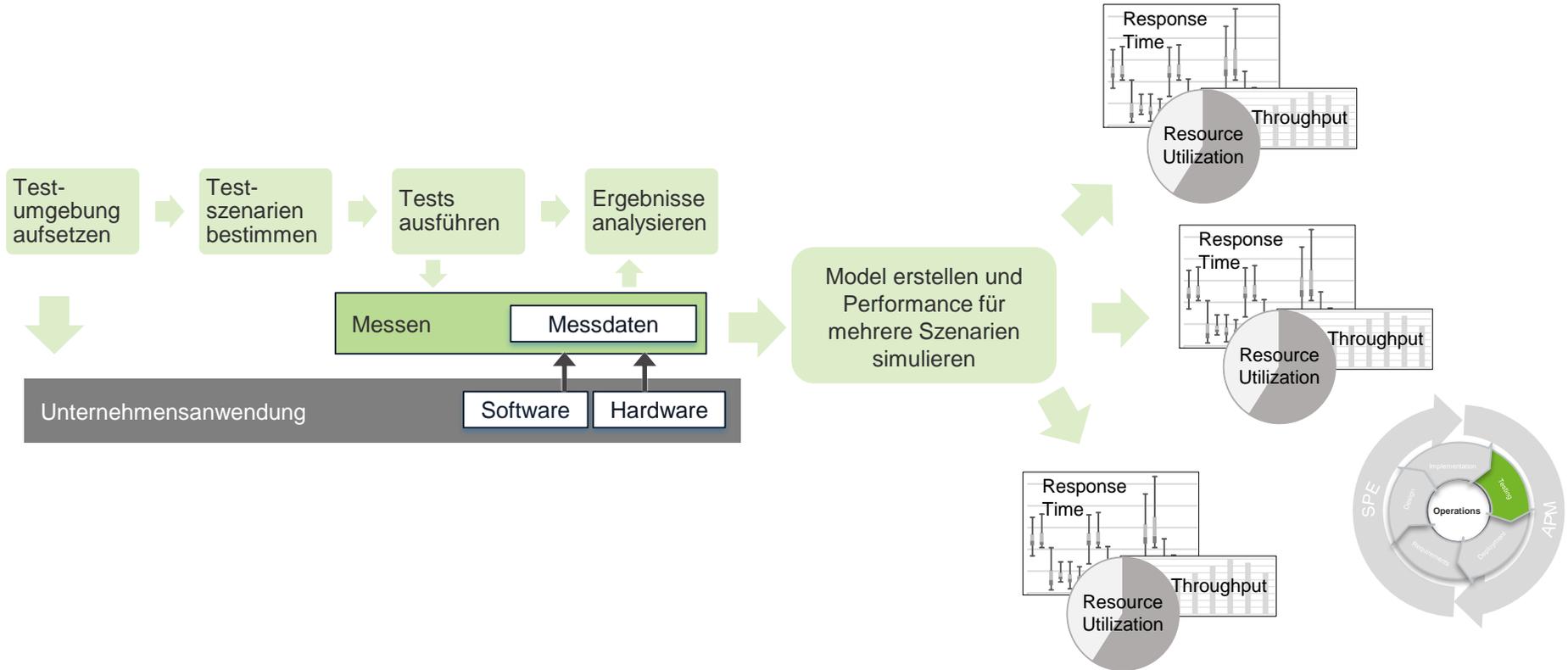
Kundenszenario – Einsatz im CI-System



- Kunde hat 3 geschäftskritische Unternehmensanwendungen(UA)
- 5 unterschiedliche Last- und Hardware Szenarien je UA
 - Mit RETIT Continuous Delivery (RCD) werden die Kosten nur im Betrieb der Testumgebungen um 250.000 € p.a. gesenkt



Anwendungsbeispiel – Lasttest-Abdeckung erweitern



Nutzen – Lasttest-Abdeckung erweitern

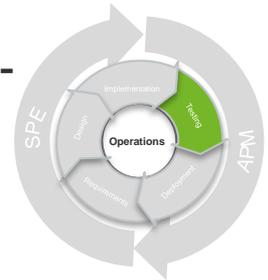
- ✓ Anzahl der Last- und Performance Tests reduzieren
→ Kosten sparen
- ✓ Testabdeckung erhöhen
- ✓ Szenarien evaluieren ohne die benötigte Hardware kaufen zu müssen
- ✓ Testabdeckung erhöhen, wenn die Anzahl der Deployments steigt



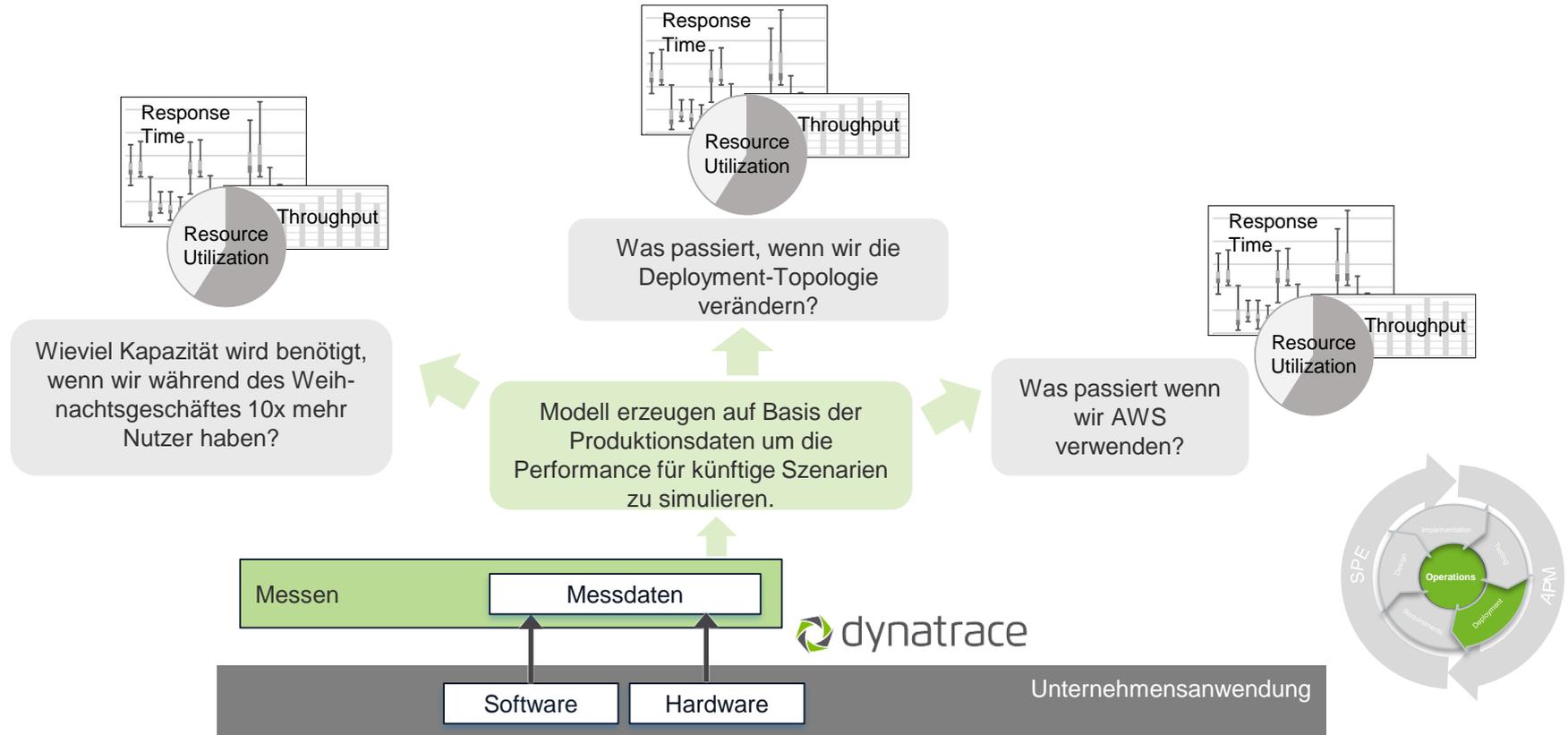
Kundenszenario – Lasttest-Abdeckung erweitern



- Ersparnis von 8-26 Personentagen (PT) wenn umfangreiche Lasttests durch kleinere Lasttests und Simulationen ersetzt werden
 - Aufwand für Lasttests (inkl. Test-Scripting, -Setup, Ausführung):
 - Klein: 13 PT, Mittel: 23 PT, Groß: 41 PT



Anwendungsbeispiel – Kapazitätsmanagement

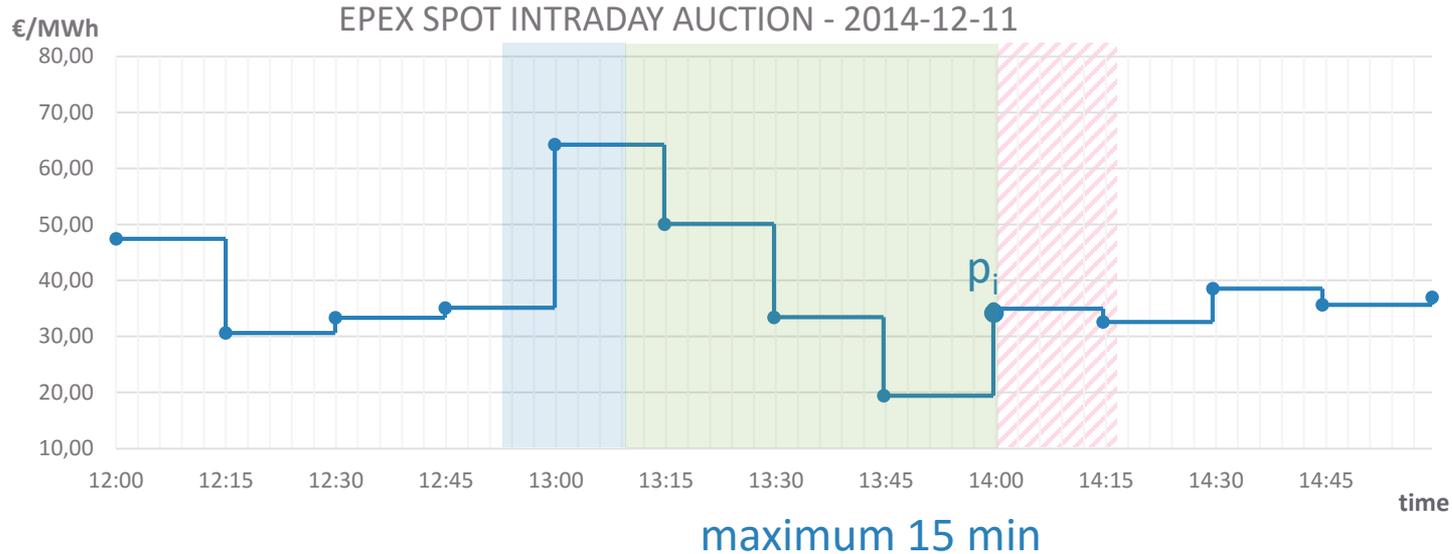


Nutzen – Modellbasiertes Kapazitätsmanagement

- ✓ **Richtiges Sizing bestimmen** – nur bezahlen was benötigt wird
- ✓ **Keine teure Testumgebung betreiben** um Änderungen zu evaluieren
- ✓ **Risiko reduzieren** z.B. Migrationen der Hardwareumgebung (Cloud)
- ✓ **Zeit beim Kapazitätsmanagement sparen**
- ✓ **Genauere Prognosen** – Mit Simulationen lineare Annahmen vermeiden



Kundenszenario – Modellbasiertes KapManagement



- Smart Grid Kapazitätsplanung für mehrere Millionen Haushalte



Agenda

Performance-Modellierung

Anwendungsbeispiele, Nutzen und
Kundenszenarien

Demo



RETIT Lösungen

- Demo



Zusammenfassung

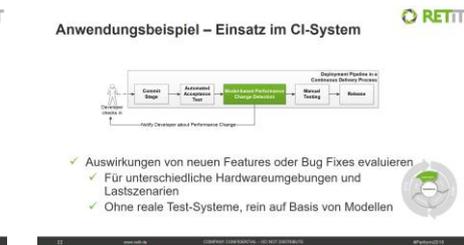
- Was haben wir heute besprochen?
- Performance-Modellierung...

✓ Verbessert die teamübergreifende Zusammenarbeit



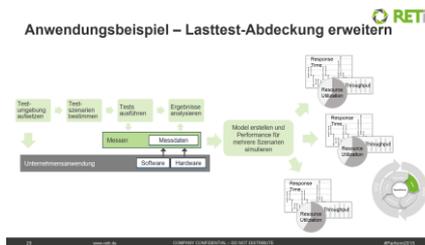
...durch die Integration zahlreicher Datenquellen

✓ Erhöht das Performance-Bewusstsein



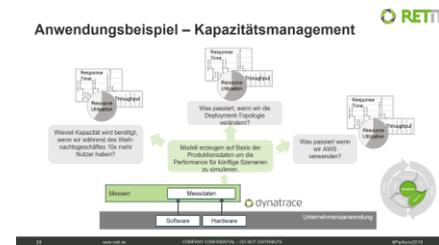
... durch direktes Feedback während der Entwicklung

✓ Erweitert die Testabdeckung



... durch die Möglichkeit, mehr Workloads und Hardwareumgebungen zu testen

✓ Spart Kosten



... da keine Mutmaßungen bei Kapazitätsplanungsaktivitäten mehr nötig sind

Vielen Dank!

Dr. Andreas Brunnert
brunnert@retit.de

RETIT

Resource Efficient Technologies & IT Systems

Peter Zahrer
peter.zahrer@dynatrace.com

