

Qualität zum Festpreis

Managed Integrations- und Test-Services für die IT-Infrastruktur einer großen Bundesbehörde

Dipl.-Math. Hermann Will

Januar 2013

Abstract: Der nachfolgende Beitrag beschreibt, wie die Integrations- und Testaktivitäten für die Applikationsplattform einer großen Behörde in Deutschland von einem externen Dienstleister in Form von Managed Services auf Festpreisbasis erfolgreich eingeführt und umgesetzt wurden. Das Thema wurde im Rahmen der Anwenderkonferenz Softwarequalität, Test und Innovation (ASQT) im September 2012 in Klagenfurt präsentiert. Der vorliegende Beitrag geht über die Präsentation hinaus und beschreibt die bis dato erzielten Ergebnisse.

1. Einleitung

Der Kunde, eine große Bundesbehörde in Deutschland, betreibt seit mehr als fünf Jahren als Plattform für seine Applikationen mehr als 500 Datenbankinstanzen auf über 200 High-End-Servern mit SPARC-Architektur und dem Betriebssystem Solaris. Inzwischen bieten neue Technologien sowohl in der Anschaffung als auch im Betrieb deutliche Kostenvorteile gegenüber der bisherigen Plattformtechnologie. Aus diesem Grund wird die bisherige Plattform durch Standard-X86-Server mit LINUX, ORACLE und weiteren Standardprodukten abgelöst.

Neben den Hardware- und Software-Komponenten für die Plattform sind weitere wesentliche Kostentreiber Installation und Konfigurierung der Plattform sowie Systemtests zur Verifizierung der Konfiguration und des korrekten Zusammenwirkens aller beteiligten Komponenten. Um auch diese Kosten unter Kontrolle zu bekommen, wird die Plattform für alle Applikationen in gleicher Weise integriert und konfiguriert, sodass ihre Bereitstellung und ihr Rollout weitgehend automatisierbar sind. Zudem werden diese Integrations- und Testaktivitäten in Form von Managed Services auf Festpreisbasis von einem externen Dienstleister durchgeführt.

Im weiteren Verlauf dieses Beitrags werden zunächst die Plattform selbst und ihr LifeCycle vorgestellt. Anschließend werden die Service Packages Systemintegration, Systemtest sowie Planung und Steuerung vorgestellt.

Eine erste Version der Plattform wurde inzwischen erfolgreich fertiggestellt und für den Applikationstest freigegeben. Die dabei erzielten Ergebnisse und die maßgeblichen Erfolgsfaktoren werden im letzten Kapitel dargestellt.

2. Plattform

2.1. Architektur

Abb. 1 zeigt die wesentlichen Komponenten der Applikationsplattform. Hardwarebasis ist ein Cluster mit zwei Standard-X86-Servern, eingebunden in die LAN¹- und SAN²-Netzwerke des Data-Centers beim Kunden. Wesentliche Software-Komponenten sind das Betriebssystem SUSE LINUX Enterprise System (SLES), ORACLE Real Application Cluster (RAC) sowie Software zur Anbindung des Systems an Data Center Verfahren wie Systemmanagement, Backup und Security.

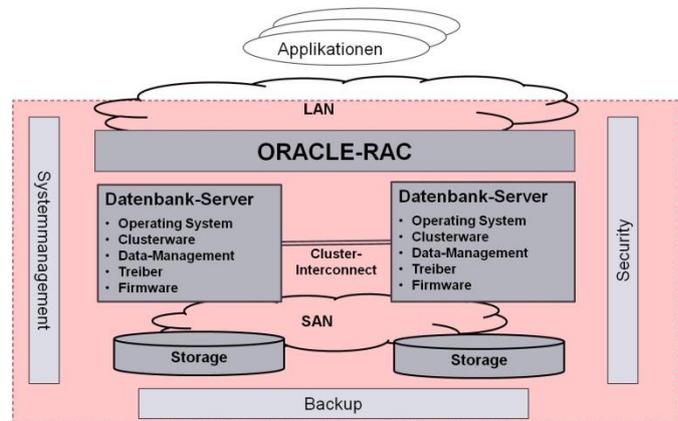


Abb. 1: Architektur der Plattform

2.2. Lifecycle

Der Kunde erwartet, dass jedes Jahr eine neue Hardware-Generation einzuführen ist, in Abb. 2 mit HW2011, HW2012, usw. bezeichnet. Die dafür erforderlichen Anpassungen werden - gebündelt mit weiteren Änderungen in der Konfiguration - als major Release SC V1.0, SC V2.0 usw. freigegeben. Diese major Releases müssen nicht nur auf der neuen Hardware-Generation ablauffähig sein, sie sollen auch vorausgegangene Releases auf den bisher schon eingeführten Hardware-Generationen ablösen. Dringende Updates zwischen den major Releases werden in Form von minor Releases freigegeben. Bevor ein solches Release in den Produktivbetrieb übernommen wird, schließen sich an die Freigabe der Plattform-Releases noch Anwender- und Applikationstests an.

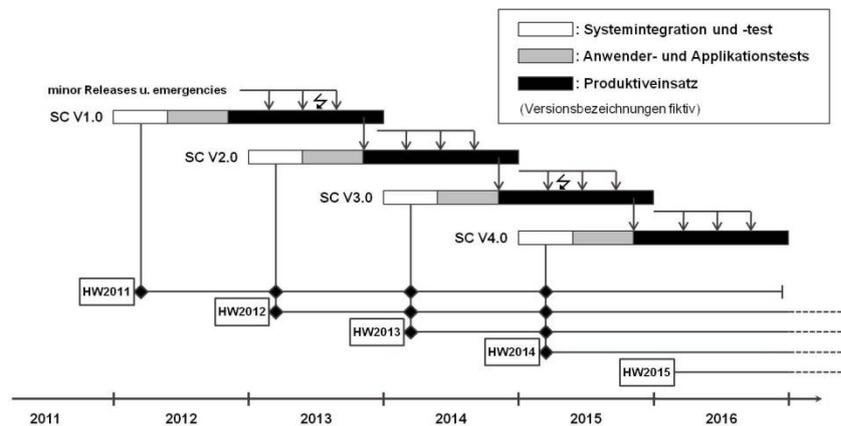


Abb. 2: Lifecycle der Plattform

¹ Local Area Network

² Storage Area Network

3. Managed Integrations- und Test Services

3.1. Systemintegration

Integration und Konfigurierung der Plattform aus ihren Einzelkomponenten wurden weitgehend mit Hilfe von Perl-, Shell- und SQL³-Skripten automatisiert, um die Plattform aufwandsarm mit hoher Qualität bereitstellen zu können, und zwar nicht nur für den nachfolgenden Systemtest sondern auch für den späteren Rollout. Die Dokumentation von Installation und Konfigurierung wurde in einem Betriebshandbuch auf Basis eines Wiki abgelegt. Der Umfang der Skripte zur Installation und Konfigurierung - im weiteren Verlauf mit I&K-Skripte bezeichnet - liegt bei ca. 3.500 Skriptzeilen. Mit diesen Skripten ist es möglich, die Installation und Konfigurierung der gesamten Software auf dem Cluster in weniger als 5 Stunden durchzuführen.

3.2. Systemtest

Der Systemtest muss sowohl die I&K-Skripte adressieren als auch das fertig konfigurierte System. Die Vorgehensweise für den Test wurde in einem Testkonzept festgelegt. Wichtige Teile in diesem Testkonzept sind Teststrategie und Testarchitektur, die in den nächsten beiden Abschnitten näher beleuchtet werden.

3.2.1. Teststrategie

Im Rahmen der Teststrategie wurden die Qualitätsziele auf Basis der in der ISO Norm 9126 formulierten Qualitätskriterien für die zwei zu betrachtenden Testobjekte, nämlich I&K-Skripte und Plattform, festgelegt und priorisiert sowie die zu ihrem Nachweis anzuwendenden Testverfahren vorgegeben.

Für die Plattform selbst haben die folgenden Ziele hohe Priorität:

- Interoperabilität:
- Alle Komponenten der Plattform wirken problemlos untereinander zusammen.
 - Um unterbrechungsfreie Upgrades und Updates zu ermöglichen, sind unterschiedliche Versionen einer Komponente miteinander verträglich.
 - Die Schnittstellen zu den Verfahren im Data-Center, wie z.B. Backup, System-Management, Auditing funktionieren ohne Probleme.
- Reife:
- Im laufenden Betrieb kommt es zu maximal einem ungeplanten Ausfall pro Quartal und Cluster mit höchstens 4 Stunden Ausfallzeit.
 - Eine Fehlerfindungseffizienz von mindestens 80% ist zu erreichen, d.h. mehr als 80% aller Fehler müssen im Systemtest, also vor den Applikationstests und dem Echteinsatz gefunden werden.
- Fehlertoleranz:
- Der Datenbank-Service bleibt bei Ausfall redundant ausgelegter Hardware-Komponenten verfügbar.
- Wiederherstellbarkeit:
- Die Wiederinbetriebnahme ausgefallener, redundant ausgelegter Hardware-Komponenten ist ohne Beeinträchtigung des laufenden Betriebs möglich.

³ Structured Query Language

- Falls Upgrades oder Updates zu Problemen führen, ist ein Rückstieg auf den Ausgangszustand ohne Beeinträchtigung des laufenden Betriebs möglich.
- Effizienz:
- Zeit- und Verbrauchsverhalten des Systems sind so gut, dass jede auf der bisherigen SPARC-Plattform betriebene Applikation mit mindestens gleicher Antwortzeit auch auf der neuen Plattform betrieben werden kann.
- Modifizierbarkeit:
- Standardänderungen an der Konfiguration sind ohne Ausfall des gesamten Clusters möglich.

Ausgehend von diesen Zielen kommen für die Plattform die folgenden Testverfahren zur Anwendung:

- Performance-Tests zum Nachweis der Performance-Ziele und zur Ermittlung der Systemeinstellungen, die erforderlich sind, um die optimale Performance für die Applikationen zu erzielen,
- Applikations-Szenarien, im Wesentlichen synthetische Datenbank-Szenarien unter den im Rahmen der Performance-Tests ermittelten Datenbank-Lasten,
- Konfigurationstests mit der Zielsetzung, dass alle Komponenten der Plattform in mindestens einem Testszenario zur Ausführung kommen,
- Administrations-Szenarien für typische Administrationsvorgänge,
- Installations- und De-Installationstests für Updates und Upgrades im laufenden Betrieb unter repräsentativer Last,
- Last- und Stresstest zur Verifizierung der Robustheit des Systems in Überlastsituationen,
- Ausfall- und Repairszenarien unter repräsentativer Last, z. B. Ausfall und Wiederinbetriebnahme eines Servers, Cluster-Interconnects oder Storage-Pfades,
- Explorative Tests im Falle von Fehlervermutungen.

Die Qualitätsziele für die I&K-Skripte weichen in Kategorie und Inhalt deutlich von den Zielen der Plattform ab, da es sich hier um eine Entwicklungsaufgabe und nicht um die Integration allgemein freigegebener Produkte und Komponenten handelt:

- Angemessenheit:
- Die Funktionalität der Skripte und ihre Dokumentation im Betriebshandbuch müssen gewährleisten, dass sie von entsprechend geschultem Personal bei der Bereitstellung für die anschließenden Applikationstests und für den Rollout verwendet werden können.
- Ordnungsmäßigkeit⁴:
- Für die Implementierung der Skripte sind nur die Bash-Shell und Perl zulässig.
 - Jedes Skript beginnt mit der Interpreterdeklaration.
 - Jedes Skript enthält einen Kommentar-Header, aus dem der Name des Skripts, der Autor, das Erstellungsdatum sowie die Funktion des Skripts entnommen werden können.
 - Jedes Skript liefert einen Exit-Status gemäß den üblichen Konventionen.
- Reife:
- Mindestens 80% Fehlerfindungseffizienz werden erzielt (s.o.).

⁴ siehe <https://www.bg.bib.de/portale/bes/Scripting/Shell/Kriterien/index.html>

- Fehlertoleranz:
 - Jedes Skript prüft, ob die erwartete Anzahl an Parametern übergeben wurde.
 - Jedes Skript prüft alle Benutzereingaben und weist falsche Eingaben oder Parameter zurück.
- Wiederherstellbarkeit:
 - Falls bei der Ausführung eines Skripts ein Fehler auftritt, muss nach Beseitigung des Fehlers das Skript ohne weitere Eingriffe erneut aufgesetzt werden können.
- Benutzbarkeit:
 - Die Anwender müssen die Skripte ohne Kenntnis des Codes bedienen können.
- Effizienz:
 - Der Automatisierungsgrad von Installation und Konfigurierung muss so hoch sein, dass ein Cluster inkl. Datenbank mit einem Aufwand von maximal fünf Stunden bereitgestellt werden kann.
- Analysierbarkeit:
 - Falls bei der Ausführung eines Skripts ein Fehler auftritt, muss eine aussagefähige Fehlermeldung ausgegeben und in einem Logfile protokolliert werden.
- Modifizierbarkeit:
 - Der Skript-Code muss entsprechend seiner Struktur nach den üblichen Regeln formatiert und eingerückt sowie ausreichend dokumentiert werden.

Zur Verifizierung dieser Ziele stehen für die I&K-Skripte die folgenden Prüf- und Testverfahren im Vordergrund:

- Reviews zur Verifizierung von Angemessenheit, Ordnungsmäßigkeit, Benutzbarkeit und Modifizierbarkeit,
- Funktionstests (Positivtests) zur Überprüfung des korrekten Ablaufs von Installation und Konfigurierung,
- Negativtests zur Verifizierung des Verhaltens bei fehlerhaften Parametereingaben,
- Fehler- und Reparaturszenarien zur Verifizierung des korrekten Verhaltens bei „äußeren Fehlern“ wie z. B. fehlende Komponente oder fehlerhafte Hardware inkl. Neustart des Vorgangs nach Beseitigung des Problems,
- Performancetests zur Verifizierung der Effizienzvorgaben.

3.2.2. Testarchitektur

Der größte Teil der Testszenarien wird im Sinne von end-to-end-Tests unter Datenbanklast durchgeführt. Für die Erzeugung der dafür erforderlichen flexiblen Lasten kommt ein sog. ORACLE-Testroboter zum Einsatz, der sich auch schon in anderen Projekten bewährt hat. Mit diesem Roboter können Lasten, z.B. auf dem Cluster-Interconnect oder auf dem Storage-System voreingestellt und automatisiert erzeugt werden. Dieser Testroboter (TR) ist die zentrale Komponente in der Testarchitektur, die in Abb. 3 skizziert ist. Die Testskripte zur automatisierten Erzeugung der Lasten und Abwicklung der Testschritte werden in der TR-Bibliothek abgelegt. Die Beschreibung der Testfälle wird in einer von der Testumgebung abgesetzten Datenbank abgelegt. Die Abwicklungsplanung der Testfälle erfolgt in Form sog. Testvorhaben. Aus diesen Testvorhaben werden Testaufträge für den Testroboter erzeugt und im TR-Repository abgelegt. Der TR-Scheduler ist für die Abwicklung der Testaufträge auf den beteiligten Instanzen und für die Ablage der Testergebnisse und -protokolle im TR-Repository zuständig.

Insgesamt wurden für die erste Version der Plattform ca. 50 Testszenarien entwickelt und abgewickelt.

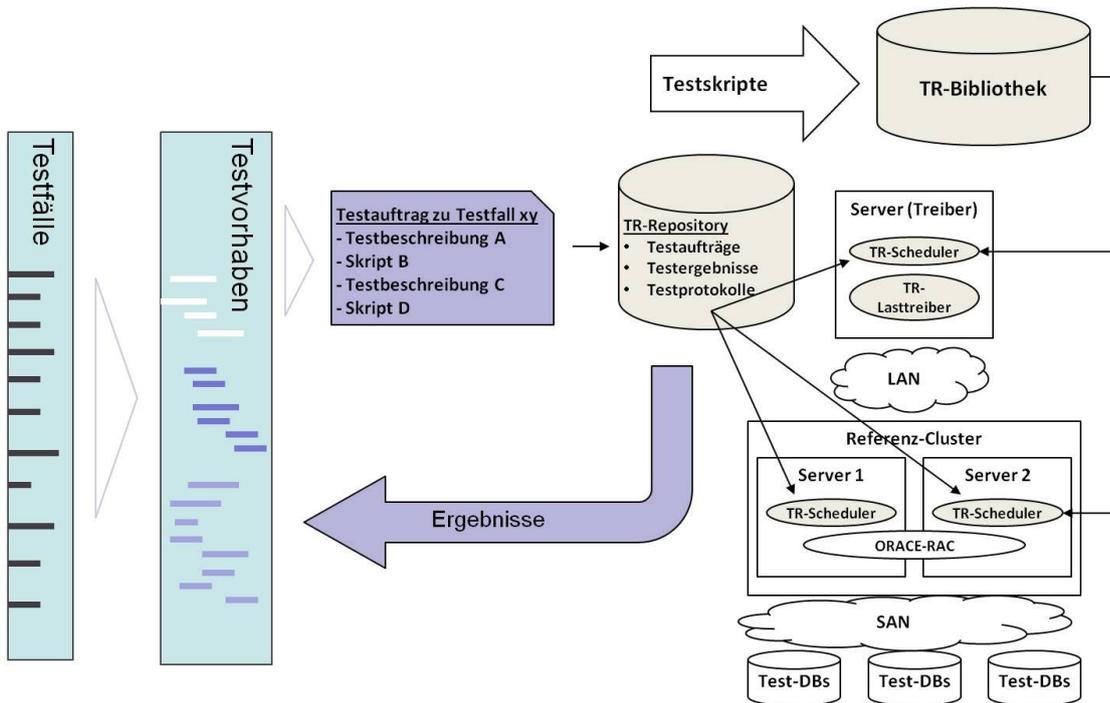


Abb. 3: Testarchitektur

3.3. Planung und Steuerung

Planung und Steuerung der Integrations- und Testaktivitäten erfolgt ebenfalls auf Basis von Service-Packages. Da Planung und Steuerung kontinuierliche Aktivitäten sind, werden diese auf Quartalsbasis vereinbart und abgerechnet. Wesentliche Aufgaben sind die inhaltliche und terminliche Planung der Plattform-Releases mit dem Kunden und mit den Lieferanten der Plattform-Bestandteile sowie die Planung und Verfolgung der Integrations- und Testaktivitäten. Letzteres erfolgt metrikbasiert, und zwar auf Grundlage von Testfall-, Problem- und Kostendaten.

4. Erfolgsfaktoren

Wie oben erwähnt, wurde inzwischen die erste Version der Plattform für den Applikationstest im vorgegebenen Termin- und Kostenrahmen freigeben. Bis zur Freigabe wurden ca. 30 Probleme gemeldet und alle schwerwiegenden Probleme korrigiert. Im Rahmen der seit ca. zwei Monaten laufenden Applikationstests wurde bislang lediglich ein Problem gemeldet, sodass zu erwarten ist, dass die vereinbarte Fehlerfindungseffizienz erreicht werden wird.

Wesentliche Faktoren für den Erfolg dieses Projekts waren und sind:

- **Kunden- und Domänen-Know-How:**
Das Projektteam war schon bei vielen anderen Projekten des Kunden beteiligt und konnte daher die für den Kunden wichtigen Testszenarien festlegen und priorisieren, sodass nicht nur richtig getestet wurde, sondern auch das Richtige getestet wurde. Neben dieser Customer Intimacy muss selbstverständlich fundiertes Wissen über alle Komponenten der Plattform und ihr Zusammenwirken im Team vorhanden sein.
- **Beteiligung des Kunden:**
Die Requirements des Kunden waren in einer Leistungsbeschreibung festgelegt. Um aber Anforderungen zu identifizieren, die seitens des Kunden als selbstverständlich erachtet werden und daher in der Leistungsbeschreibung fehlen, war die Einbindung des Kunden in Form von Reviews, regelmäßigen Status-Meetings und Vorführung von Zwischenständen von großer Bedeutung für den Projekterfolg.
- **Ergebnisfokussierung:**
In diesem Zusammenhang sind zwei Aspekte von Bedeutung: Zum einen war es wichtig, dass die Ergebnisse der Packages in der Leistungsbeschreibung klar, d.h. überprüfbar und messbar, vorgegeben waren. Zum anderen ergab sich durch die Strukturierung des Projekts in Packages der Zwang, die Packages termingerecht und vollständig abzuschließen und so das in vergleichbaren Projekten häufig anzutreffende 90%-Syndrom zu vermeiden.
- **Risikomanagement:**
Um die durch die fixen Preis- und Terminfestlegungen für die Packages gegebenen Risiken zu managen, wurden auf Basis des Kunden- und Domänen-Know-Hows Testfälle priorisiert, Aktivitäten wo möglich parallelisiert - z.B. Testentwicklung und -abwicklung - und durch frühzeitige Smoke⁵-Tests verifiziert, dass die Plattform tragfähig ist.

⁵ Testszenarien, die die Hauptfunktionalität eines Systems abdecken mit dem Ziel, festzustellen, ob die wichtigsten Funktionen des Systems arbeiten.



Qualität zum Festpreis Managed Integrations- und Test-Services für die IT-Infrastruktur einer großen Bundesbehörde

Hermann Will, 06.09.2012

hermann.will@gadvice.de

Agenda

Der Kunde
Die Anforderungen
Die Umsetzung

Lt. IBM schlucken Installation und Wartung der **Computer-Infrastruktur** derzeit **70 Prozent der IT-Budgets** in Unternehmen
([heise News, 12.04.2012](#))

Durch **Verbindungsprobleme** verursachte Störungen der EC2 Cloud Services von Amazon führt zu Verlusten von Kundendaten.
([heise News, 28.04.2011](#))

Twitter Ausfall am 26.07.2012 war "**doppeltes Pech mit der Infrastruktur**". In den Rechenzentren hat die Redundanz nicht so gewirkt wie eigentlich gedacht. Der Ausfall eines Systems konnte nicht durch ein Parallelsystem aufgefangen werden, da dies nahezu gleichzeitig ausgefallen sei.
([heise News, 26.07.2012](#))

3 Stunden lang konnten auf den Bürgerämtern in Berlin keine Ausweise und Pässe beantragt sowie Autozulassungen vorgenommen werden, da ein **Server-Update fehlschlug**.
([Berliner Zeitung, 13.04.2012](#))

Eine **Computerpanne bei der Berliner S-Bahn** führt zu zahlreichen Zugausfällen und massiven Verspätungen im Berufsverkehr. Nach Wartungsarbeiten konnte ein zentrales Stellwerk in Halensee gegen 04.00 Uhr nicht in Betrieb genommen werden und die Verkehrslenkung musste per Hand vorgenommen werden.
([DIE WELT, 21.05.2012](#))



Der Kunde

06.09.2012

Copyright Hermann Will 2012

Folie 3

Profil

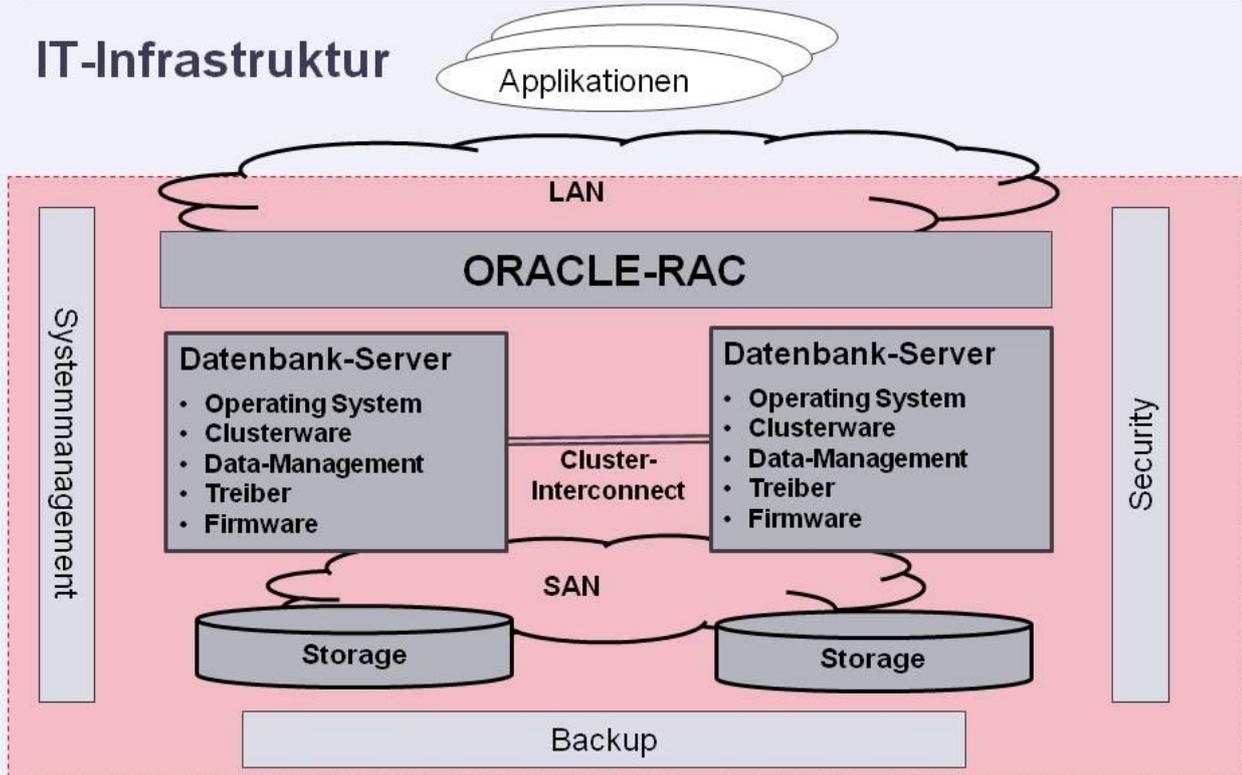
- Körperschaft des öffentlichen Rechts
- Zentrale in Nürnberg
- 10 Regionaldirektionen
- 178 Agenturen
- ca. 600 Geschäftsstellen
- 113.000 Beschäftigte
- ca. 220 High-End-Server
- mehr als 200 IT-Verfahren

06.09.2012

Copyright Hermann Will 2012

Folie 4

IT-Infrastruktur



06.09.2012

Copyright Hermann Will 2012

Folie 5

Die Anforderungen

06.09.2012

Copyright Hermann Will 2012

Folie 6

Managed Services.....

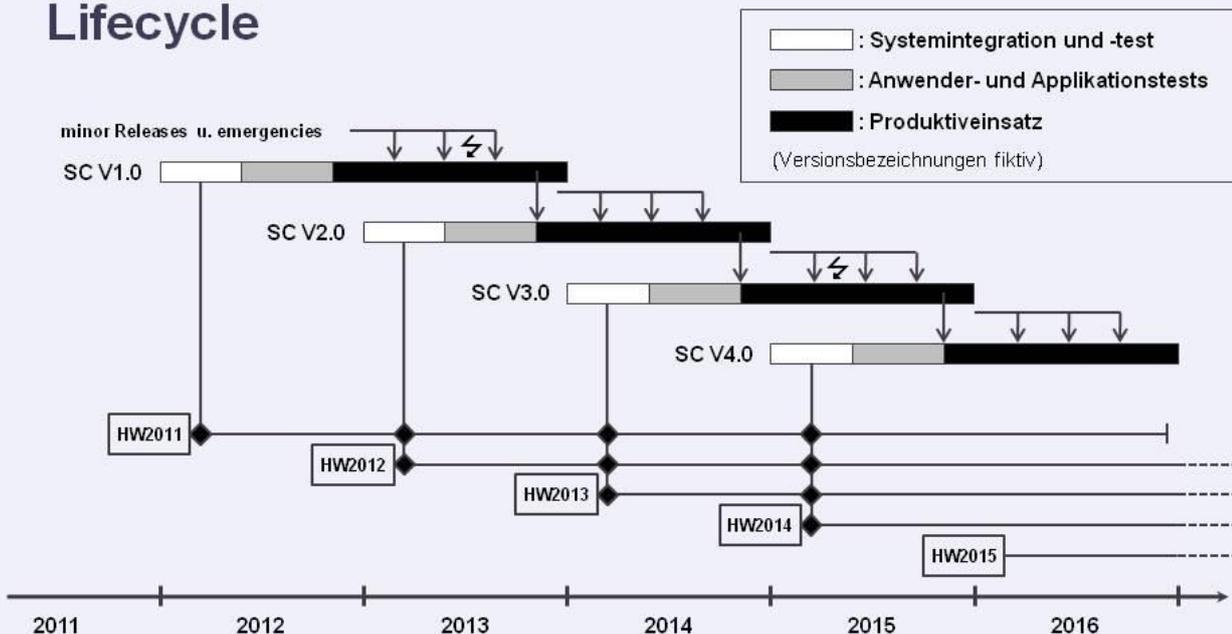


..... sind IT Service-Leistungen, die von einem spezialisierten Anbieter für einen fest definierten Zeitraum bereitgestellt werden. Die im Vorfeld definierten Leistungen können vom Kunden zu jeder Zeit nach Bedarf abgerufen oder abbestellt werden.

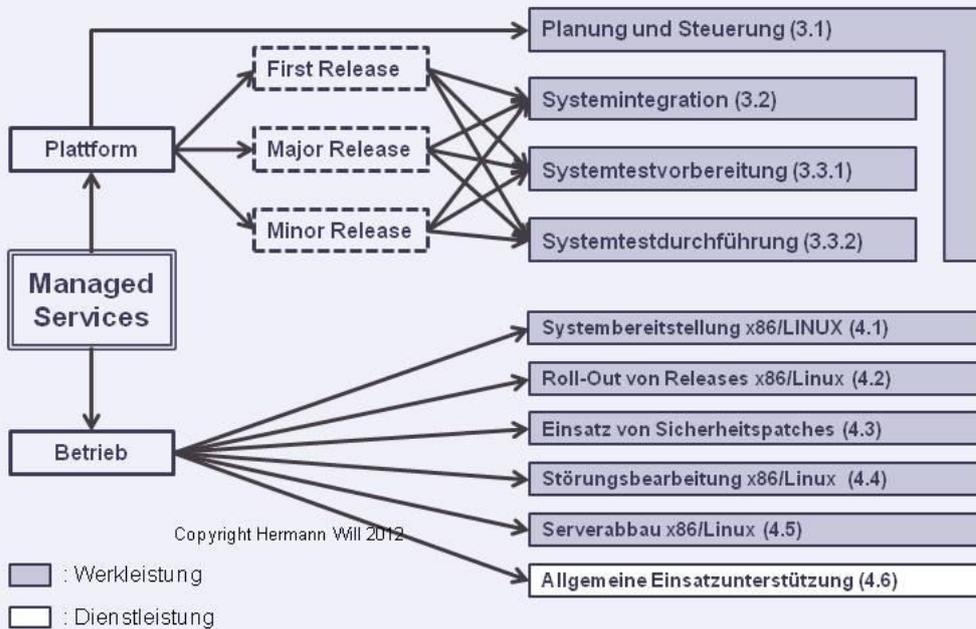
ITIL®-Glossar

(ITIL: IT-Infrastructure Library)

Hardware- und Software-Lifecycle



Managed Service Packages



06.09.2012

Folie 9

Die Umsetzung

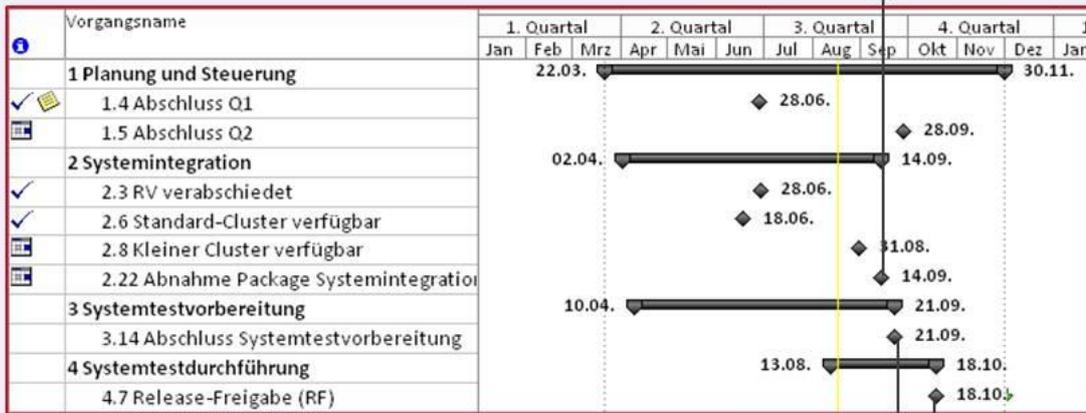
06.09.2012

Copyright Hermann Will 2012

Folie 10

Makroplan 1st Release

- Release-Vereinbarung
- I&K-Leitfaden
- Installationsset
- Configuration Verificaton Tests



- Testhandbuch
- Testfälle
- Testumgebung
- Testbericht
- Testprotokolle
- Release-Info

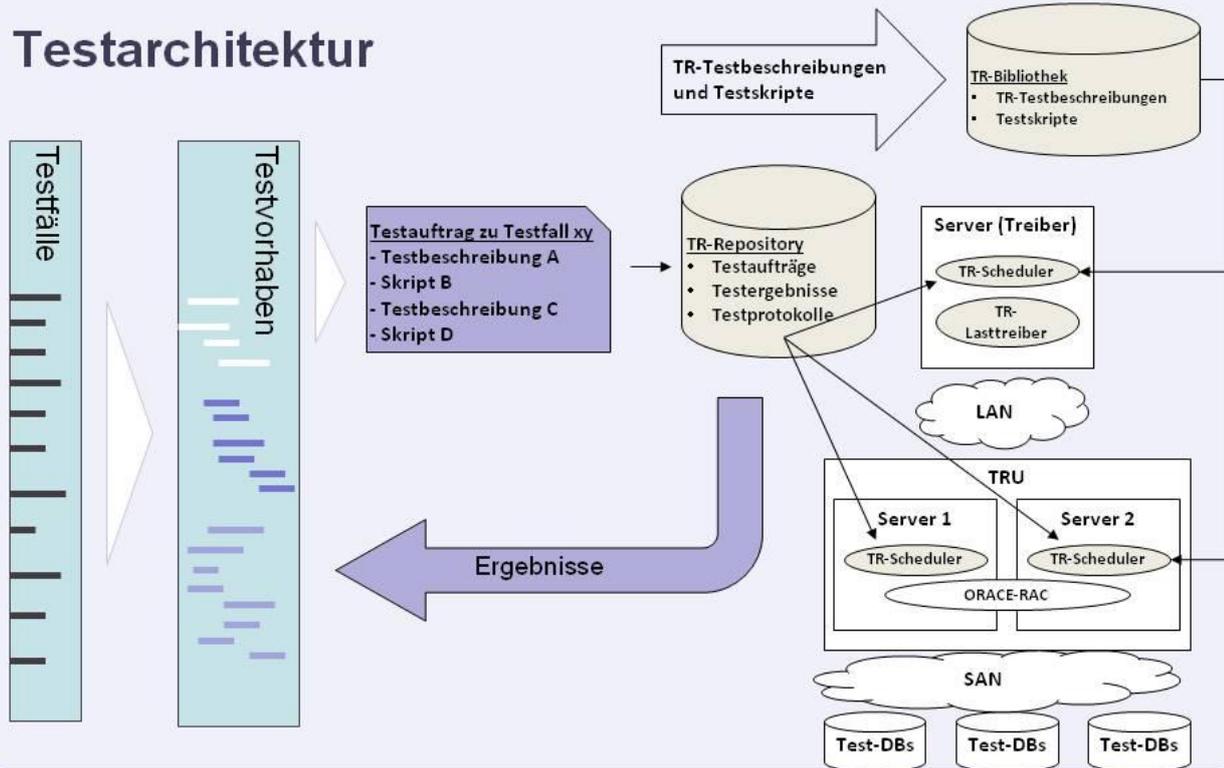
Prüf- und Teststrategie

Prüfobjekte	Installations- und Konfigurationsskripte	Applikationsplattform
Qualitäts-Merkmale im Fokus	<ul style="list-style-type: none"> • Ordnungsmäßigkeit • Benutzbarkeit • Effizienz • Analysierbarkeit und Modifizierbarkeit 	<ul style="list-style-type: none"> • Interoperabilität • Reife • Fehlertoleranz und Wiederherstellbarkeit • Effizienz • Modifizierbarkeit
Prüfverfahren	<ul style="list-style-type: none"> • Reviews • Funktionstests inkl. Negativtests • Fehler- und Restart-Szenarien • Usability-Tests 	<ul style="list-style-type: none"> • Performance-Tests • Applikations-Szenarien • Konfigurationstests • Administrations-Szenarien • Last- und Stresstest • Ausfall- und Repairszenarien

Systembereitstellung in 8h

99,8% Verfügbarkeit

Testarchitektur



06.09.2012

Copyright Hermann Will 2012

Folie 13



06.09.2012

Copyright Hermann Will 2012

Folie 14