Continuous Delivery in der Praxis

Big Techday 5

15.06.2012

München

Was ist Continuous Delivery?

> Continuous Delivery beschreibt Konzepte und Praktiken mit denen häufige und stressfreie Releases möglich werden. Continuous Delivery geht dabei über Continuous Integration und Continuous Deployment hinaus. Es umfasst agiles Testen, hochgradige Automatisierung, cross-funktionale Teams (Entwickler, Tester, Administratoren) und Lean Thinking.
>
> www.continuousdelivery.de

Continuous Delivery in der Praxis

1. Das WITA Projekt
2. Ausgangssituation
3. Deployment-Automatisierung
4. Akzeptanztests
5. Konformitätstest
6. Produktions-Deployment
7. Mehrere Teams
8. Go-Live

Das WITA Projekt

Das WITA Projekt

- Initiiert von M-net Telekommunikations GmbH
  - Gegründet 1996
  - Gesellschafter:

SW//M

Stadtwerke Augsburg

AÜW online
Strom für das Allgäu

N-ERGIE

ESTW
ERLANGER STADTWERKE
... nichts liegt näher

mehr als Energie
infra

  - Mitarbeiterstand: 700
  - Umsatz 2011: ca. 179 Mio. Euro

Das WITA Projekt

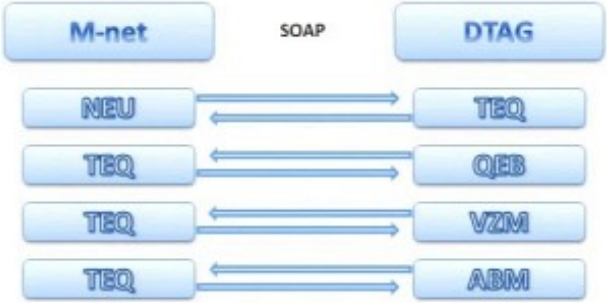
- Wholesale IT Architecture
- Orderschnittstelle der DTAG
  - Bestellung von TALs
  - Bestellung von Reselling-Produkten (xDSL)
- Asynchrone Webservice-Schnittstelle (SOAP)
- Nachfolger von ESAA (File-basierte Schnittstelle)

Das WITA Projekt

- Beispiel einer Neubestellung

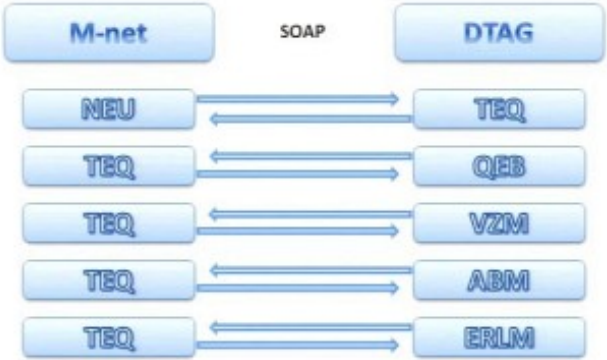
Das WITA Projekt

- Beispiel einer Neubestellung



Das WITA Projekt

- Beispiel einer Neubestellung



Das WITA Projekt

- Beispiel einer Neubestellung

Das WITA Projekt - Ziele

- Ablösung der ESAA / Go-Live WITA
  - Weniger TAL-Bestellungen per FAX
  - Schnellere Rückmeldungen der DTAG
  - Bessere Status-Übersicht
- Konformitätstest (KFT) bestehen
- SLAs einhalten
- Hohe Testabdeckung
- Schnelle Reaktion auf Fehler / unerwartetes Verhalten

Das WITA Projekt - Ziele

- Ablösung der ESAA / Go-Live WITA
  - Weniger TAL-Bestellungen per FAX
  - Schnellere Rückmeldungen der DTAG
  - Bessere Status-Übersicht
- Konformitätstest (KFT) bestehen
- SLAs einhalten
- Hohe Testabdeckung
- Schnelle Reaktion auf Fehler / unerwartetes Verhalten

Das WITA Projekt - Zusammensetzung

- Gemischtes Projektteam

- Zusammensetzung
  - 5 Entwickler
  - 1 Betrieb & Support
  - 1 Fachbereichskoordinator
  - 1 ScrumMaster, 1 ProductOwner

Das WITA Projekt - Zeitraum

- Start: 03.05.2011
- KFT: 16.08.2011 - 24.08.2011
- RampUp Phase: 31.10.2011
- Go-Live: 14.11.2011
- Projektende: 30.03.2012

Das WITA Projekt - Zeitraum

- Start: 03.05.2011
- KFT: 16.08.2011 - 24.08.2011
- RampUp Phase: 31.10.2011
- Go-Live: 14.11.2011
- Projektende: 30.03.2012

Ausgangssituation

Ausgangssituation - Systemlandschaft

Ausgangssituation - Systemlandschaft

- Systemlandschaft ESAA

Ausgangssituation - Systemlandschaft

- Systemlandschaft ESAA

- Systemlandschaft WITA

Ausgangssituation - Builds und Deployment

- CI Builds bei M-net vorhanden
  - Via Jenkins (mit ANT & Ivy, Artifactory)
    - Webbasiertes System für CIs
    - Ausführung von Jobs
    - Anbindung an Versionsverwaltungssysteme
- Deployment Test-Systeme automatisiert
- Deployment Produktions-Systeme manuell

Ausgangssituation - Builds und Deployment

- CI Builds bei M-net vorhanden
  - Via Jenkins (mit ANT & Ivy, Artifactory)
    - Webbasiertes System für CIs
    - Ausführung von Jobs
    - Anbindung an Versionsverwaltungssysteme
- Deployment Test-Systeme automatisiert
- Deployment Produktions-Systeme manuell

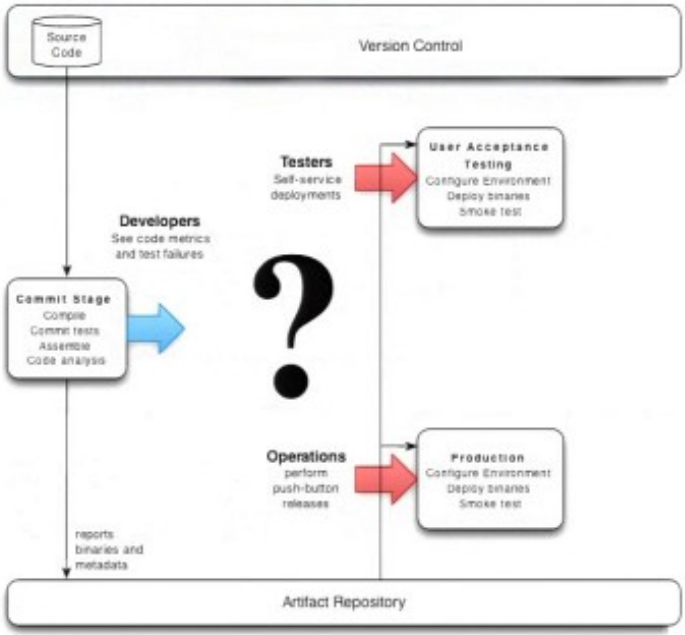


Diagramm ähnlich zu Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation (Addison-Wesley Signature)

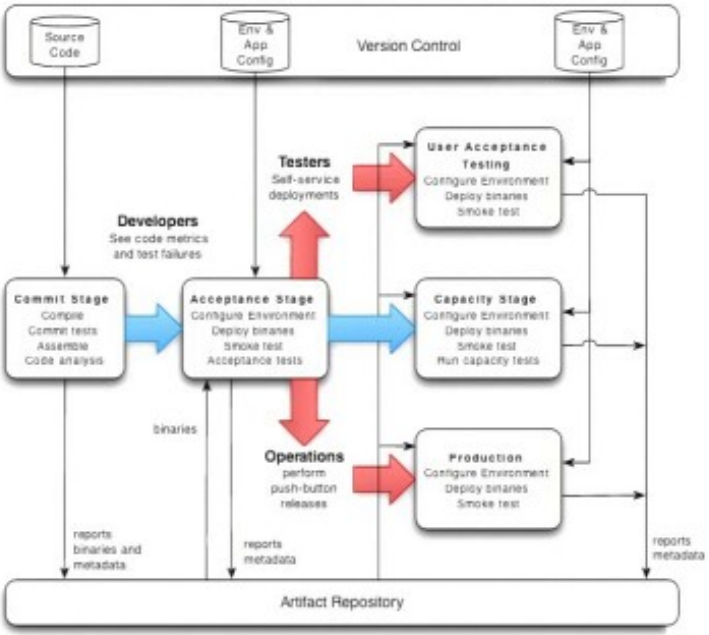


Diagramm ähnlich zu Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation (Addison-Wesley Signature)

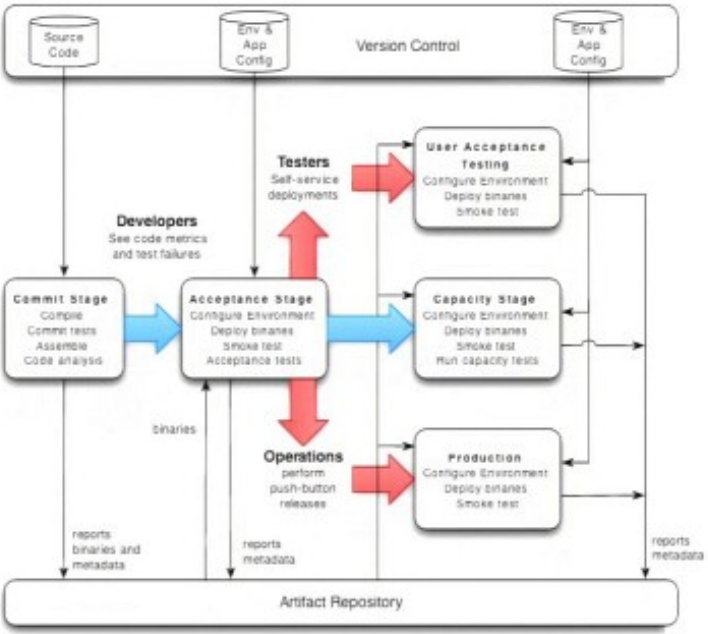


Diagramm ähnlich zu Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation (Addison-Wesley Signature)

Ausgangssituation - Herausforderungen

- Serverlandschaft wesentlich umfangreicher
- Automatisierte E2E-Tests
- Server-Konfiguration zur Build-Zeit
  - Langsame Feedbackzyklen bei Konfigurationsfehlern
  - Welche Konfiguration ist aktiv?
- Eigener Deployment Job pro Server

Deployment - reloaded

Deployment - reloaded

Für die neuen Applikationen

- Konfiguration via Templates und Properties
- Konfiguration beim Deployment
- Von 1 -> 5+ produktionsähnlichen Umgebungen

Für die neuen Applikationen

- Via Jenkins & Slaves
- 1 Jenkins-Job für alle Umgebungen
  - Parameterized Trigger
  - Node Label Parameter Plugin

Tooling

- Ant für Skripting
- Ivy für Artefakt-Management
- Artifactory als Artefakt-Repository
- Jenkins für
  - Orchestrierung
  - Remoting
  - Reporting

Tooling

- Ant für Skripting
- Ivy für Artefakt-Management
- Artifactory als Artefakt-Repository
- Jenkins für
  - Orchestrierung
  - Remoting
  - Reporting

Configuration-Management

- Property-Files
  - Hierarchisches Laden
  - Viele Properties
  - Verwaltung im SVN

```
webapp.install.path=C\:\\Temp\\wita-fassade
webapp.port.prefix=480
webapp.http.port=${webapp.port.prefix}80
webapp.service.name=tomcat-wita_${webapp.http.port}
webapp.startup.timeout=60
tomcat.java.opts=-XX:MaxPermSize=128m -Xmx256m

activemq.install.path=C\:\\Temp\\activemq
activemq.snmp.port=51510
activemq.broker.port=51515
activemq.admin.port=8161
activemq.master.slave.config=
activemq.redelivery.max.redeliveries=3
activemq.redelivery.initial.redelivery.delay=1000
activemq.redelivery.redelivery.delay=1000
...
```

Akzeptanztests

Akzeptanztests

- Alle KFT-Tests müssen automatisiert sein
- Mit deployten Applikationen
- Umgebung produktionsähnlich

```
accTestWorkflow.sendBereitstellung(builder)
               .waitForTEQ()
               .waitForQEB()
               .waitForABM()
               .waitForERLM()
               .waitForENTM();
```

Akzeptanztests

- Alle KFT-Tests müssen automatisiert sein
- Mit deployten Applikationen
- Umgebung produktionsähnlich

```
accTestWorkflow.sendBereitstellung(builder)
               .waitForTEQ()
               .waitForQEB()
               .waitForABM()
               .waitForERLM()
               .waitForENTM();
```

Akzeptanztests

- Telekom durch Simulator ersetzt
- Antwortreihenfolge konfigurierbar
  - z.B. QEB, ABM, ERLM, ENTM
- Setup durch Attribut der Nachricht ausgewählt
- Rückmeldungen in konfigurierbarem Abstand

Herausforderungen

- Laufzeit
  - Spätes Feedback
  - Sleep ist schlecht

Herausforderungen

- Laufzeit
  - Spätes Feedback
  - Sleep ist schlecht

Lösung

- Polling
- Parallelität
- Ergebnis:
  - Laufzeit von > 1h → 15 Minuten
  - Versteckte Multi-Threading Probleme gefunden

Alle Tests kaputt!

Nach 2 Stunden Laufzeit!

Asynchrone Tests

- Laufen in Timeout
- > 2 Minuten
- Dadurch E2E-Tests sehr langsam
- Feedback zu spät!

Deployment "kaputt"

- Container hochgefahren
- Anwendung kaputt
- Meist Konfiguration falsch

Frag einfach!

Status Servlet

- Einfache Website
- Auskunft über verschiedene Teile der Anwendung
- Automatisch abfragbar

Status Servlet

- Abfrage beim Deployment
- Deployment schlägt fehl, wenn Status nicht grün

Konformitätstest (KFT)

Konformitätstest (KFT)

- Voraussetzung für Go-Live der WITA Schnittstelle
- Genau definierte Use-Cases und Testdaten
- Durchführung mit spezieller DTAG Testumgebung
- Ziel: positives Testat von DTAG

Konformitätstest (KFT) - Vorgaben DTAG

- Maximale Inaktivität während KFT: 2 Wochen
- Im Ø nur 3 Fehlversuche pro Testfall erlaubt
- Erfolgreiche Tests nicht wiederholen
- Testabbruch durch DTAG möglich

Konformitätstest (KFT) - Projektauswirkungen

- KFT mit produktionsähnlicher Umgebung
- Schnelles Re-Deployment der KFT Umgebung notwendig
- Testfälle automatisiert
  - Einzeln per Jenkins-Job ausführbar
  - Automatisch nach jedem Check-In (E2E-Test)

Konformitätstest (KFT) - erfolgreich

Datum	durchgeführt	%	positiv	%	Gesamt
16.08.2011	16	27,12 %	1	6,25 %	1,69 %
17.08.2011	28	47,46 %	4	14,29 %	6,78 %
18.08.2011	34	57,63 %	28	82,35 %	47,46 %
19.08.2011	52	88,14 %	49	94,23 %	83,05 %
22.08.2011	56	94,92 %	55	98,21 %	93,22 %
23.08.2011	59	100 %	56	94,92 %	94,92 %
24.08.2011	59	100 %	59	100 %	100 %

Produktions-Deployment

Produktions-Deployment

- KFT Umgebung ist produktionsähnlich
- Deployment sollte dort so laufen wie auf Produktion
- Wird oft stattfinden → automatisieren

Produktions-Deployment

- In enger Zusammenarbeit mit IT-BS (im Team)
- Am richtigen Ort (/opt/app/tomcat-...)
- Rechte anpassen
- Init-Skript
- Auf eigenem Jenkins
- Per Ant-Skript

Am richtigen Ort

- Vor Deployment alles löschen
- Dann Daten entsprechend entpacken
- Benutzt sudo mit Wildcards

Rechte anpassen

- Eigenes Skript-Template
- Via sudo ausgeführt
- Kommt direkt von Betrieb

Init-Skript

- Via Templating
- In /etc/init.d
- Wird bei Server Neustart automatisch gestartet
- Überprüft Status Servlet

Init-Skript

- Via Templating
- In /etc/init.d
- Wird bei Server Neustart automatisch gestartet
- Überprüft Status Servlet

Produktions-Jenkins

- Eigener Jenkins
  - Entwickler nur eingeschränkten Zugriff
  - Produktions-Server als Slaves
  - Nur Master kann auf Artifactory & SVN zugreifen
- Deployment zweigeteilt
  - Checkout und Artefakt-Download
  - Installation und Startup

Mehrere Teams

Mehrere Teams

- Arbeiten an der gleichen Software
  - WITA Projekt
  - Linienentwicklung
- Auf verschieden Branches
  - Verschiedene Release-Zyklen
  - Angst
- Herausforderungen
  - CI
  - Deployment

Mehrere Teams

- Arbeiten an der gleichen Software
  - WITA Projekt
  - Linienentwicklung
- Auf verschieden Branches
  - Verschiedene Release-Zyklen
  - Angst
- Herausforderungen
  - CI
  - Deployment

SVN Post Commit Hook

- Sucht nach autobuild.yaml
- Löst Jenkins-Jobs mit Parametern aus
- SVN-Branch ist immer ein Parameter
- Jenkins-Job wird auf jeweiligem Branch ausgeführt

```
jobs_to_run:
    WITA_Fassade_CI:
        default:
            ENVIRONMENT: ci
            INTEGRATION_TEST_POSTFIX:
        branches/RELEASE:
            ENVIRONMENT: cir10
            INTEGRATION_TEST_POSTFIX: -release
```

Probleme

- Release Build hängt in Queue
  - Evtl. Lösung: Jenkins Plugin?

Go-Live

Go-Live

- Wenn Bestellung nicht funktioniert, dann steht das Geschäft
- Parallelbetrieb
  - Wenige Key-User arbeiten bereits mit WITA
  - Rest mit ESAA
  - Unterscheidung nur in GUI
  - Server kann beides
  - Start-Parameter definiert ESAA / WITA Modus

Go-Live

- Nach 2 Wochen genug Vertrauen in Schnittstelle
- Dann alle auf WITA
- Teilweise mehrere Releases pro Woche
  - Bugfixes
  - Reaktionen auf Verhalten der DTAG
  - Neue Features

Post Go-Live

- Releases alle 2 Wochen nach Sprintende
- Applikationsbetreuung im Team zur Übergabe
- Langsames Auslaufen des Projekts

Post Go-Live

- Releases alle 2 Wochen nach Sprintende
- Applikationsbetreuung im Team zur Übergabe
- Langsames Auslaufen des Projekts

Joachim Glink
M-net Telekommunikations GmbH
joachim.glink@m-net.de

Dr. Stefan Wolf
TNG Technology Consulting GmbH
stefan.wolf@tngtech.com

Danke für Ihre Aufmerksamkeit!