

The logo for ZiM (Zentrum für Information Management) features the letters 'ZiM' in a bold, blue, sans-serif font. The 'i' is lowercase and has a dot. The background is a dark blue gradient with several light blue speech bubbles of various sizes and orientations.The logo for 'Talk' features the word 'Talk' in a large, bold, red, italicized sans-serif font. Below it, the tagline 'WISSEN SCHAFFT IT' is written in a smaller, red, all-caps sans-serif font. The logo is set against a white speech bubble background.

Devops & QA in der Praxis – Ein Vorgehensmodell mit Foreman, Puppet und Selenium

UNIVERSITÄT
DUISBURG
ESSEN

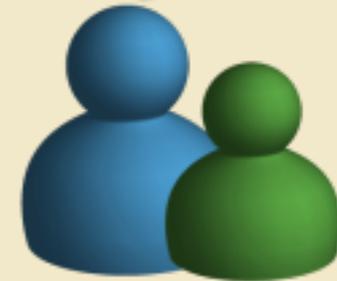
Offen im Denken

 Daniel Biella / Malte Hermsen ■ 27.03.2015



„Sure glad the hole isn't at our end“

User: „...?“



- **Früher**

- „Die Software erfüllt die Spezifikation“

- **Heute**

- Agile Entwicklungsmethoden
 - Fokus auf „Mehrwert“ der Software für Nutzer
 - Veränderungen im Zyklus Entwicklung/Qualitätssicherung
- Cloud
 - Software wird zum Service, und...

Ein Service erfüllt Aufgaben aus Nutzersicht

Qualitätsmaß: Wie nützlich ist der Service?

- **Operabilität**: Betriebsfähigkeit, Handhabbarkeit, Bedienbarkeit, Usability
- **Funktionalität**: „Tut, was es soll“, Liefert gewünschte Ergebnisse
- **Verfügbarkeit**: Service ist für alle Benutzergruppen wohl definiert & verfügbar
- **Kohärenz**: Integration von Drittsystemen, Schnittstellen, Stabilität bzgl. Updates und Upgrades



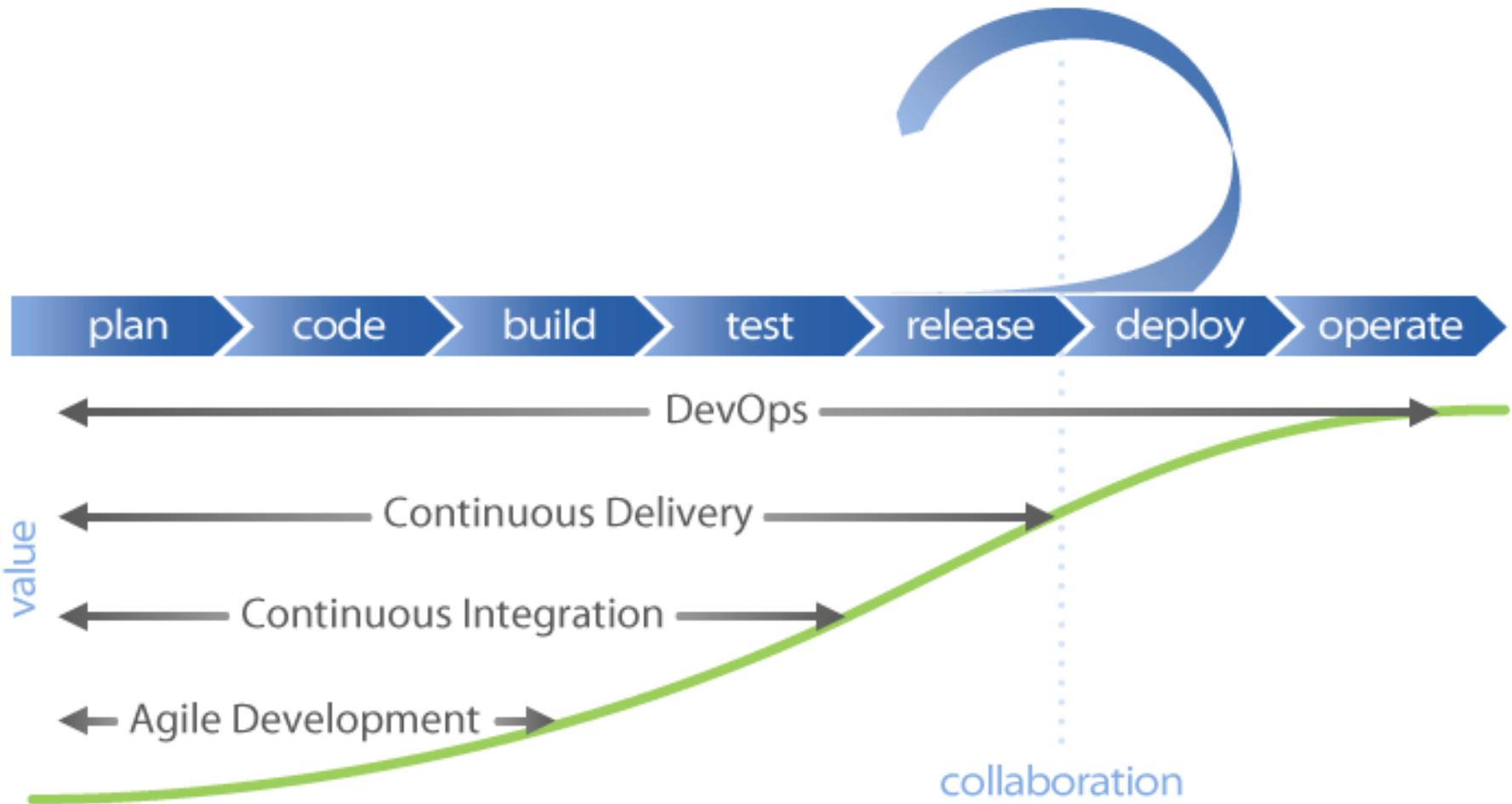
- **Überblick über DevOps & QA**
- **Foreman und Puppet**
 - Hands-on
- **QA in der Praxis**
 - Selenium
 - Hands-on
- **Fazit**

- **Betreiberanforderungen zunehmend dynamisch**
- **Nutzerverhalten zunehmend dynamisch**
- **Forderung nach geringer *time-to-market***
- **Hohe Update- bzw. Releasezyklen**

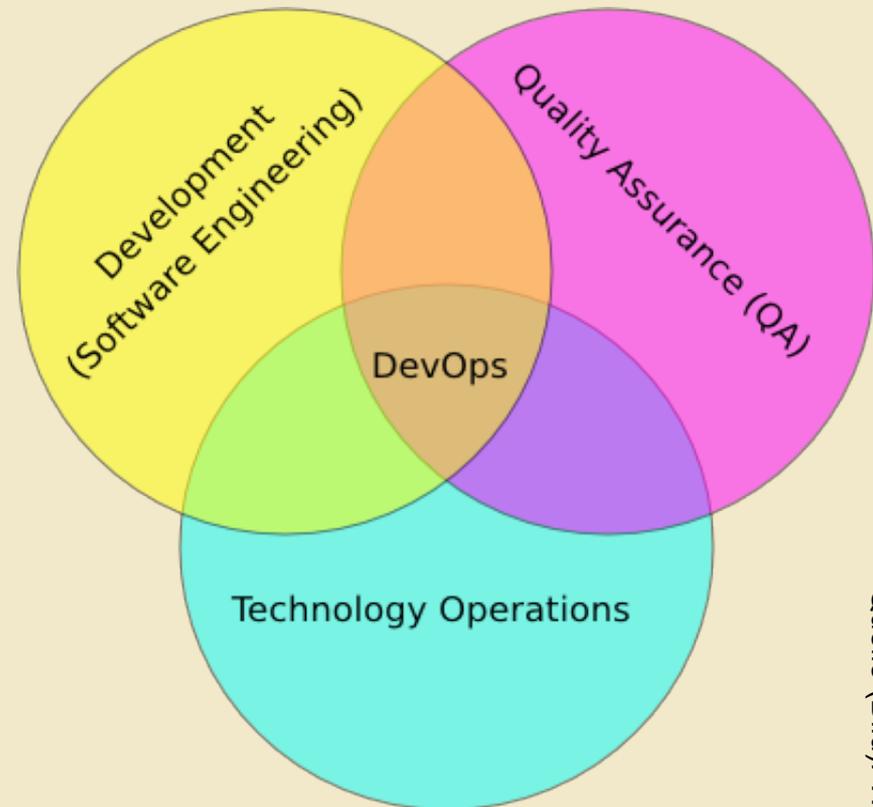
- **ITIL: Hoher Standardisierungsgrad, aber nicht zwingend agil**
- **Methoden aus der Software-Entwicklung:**
 - Continuous Integration (CI)
 - Continuous Delivery (CD)

- Wortkreuzung aus „Development“ und „Operations“
- Definition uneinheitlich, Beispiele (Auszug):
- “...DevOps – essentially, bringing the development and operational teams closer together through **automation** to support the speed of change the business requires.” (Clive Longbottom, ComputerWeekly.com)
- “...DevOps is “a cross-disciplinary community of practice dedicated to the study of **building, evolving and operating** rapidly-changing resilient **systems at scale**.” (Jez Humble, Agile Admin Blog)
- “The primary goal of DevOps is making **reliably performing production software** that can be changed rapidly.” (Charles Babcock, InformationWeek)

DevOps vs CD vs CI vs AD



- **Ursprünge in agilen Software-Entwicklungsmethoden**
- **Übernahme agiler Methodik in IT Operations und Service Delivery**
- **„Erweitere das agile Prinzip auf die Infrastruktur“**
- **Fokus auf Automatisierung**



• Entwicklung

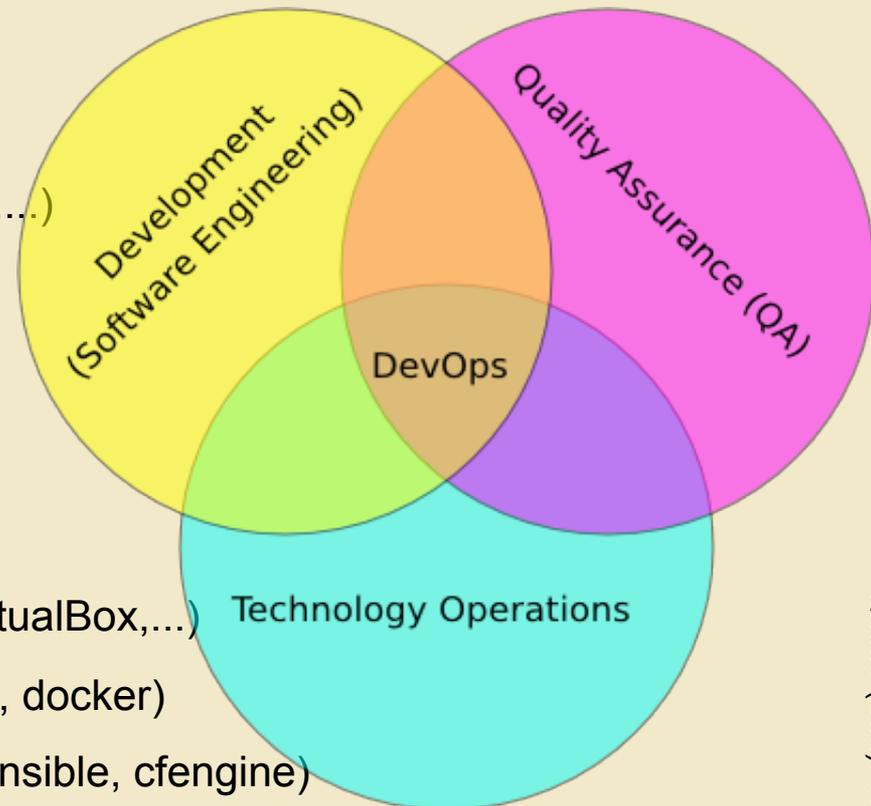
- Automatisierte Release-/Build-Tools (z.B. jenkins, travisci,...)
- Versionskontrolle (z.B. git, svn, cvs)

• Qualitätssicherung

- Automatisierte Tests (z.B. Selenium, Jmeter,...)
- Herausforderung: Testfallerstellung

• IT Operations

- Monitoring/Metriken (statsd, u.v.m.)
- Orchestrierung (zookeeper, noah, mesos)
- Virtualisierung (VMWare, Hyper-V, KVM, VirtualBox,...)
- Containerization (AWS, OpenStack, vagrant, docker)
- Konfigurationsmanagement (puppet, chef, ansible, cfengine)



DevOps → Entwicklung & IT-Betrieb...

Ein mögliches Vorgehensmodell

Foreman, puppet



git



Selenium



Betriebssystem
Middleware
Framework
Tool
Paket



Versionierte
Anwendung



Versionierte
Anwendungs-
Konfiguration



Versionierte
SAF



IaaS/PaaS

SaaS

Qualitäts-
sicherung

...mit Foreman und Puppet

Foreman – Login

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken



Benutzername

Passwort

Anmeldung

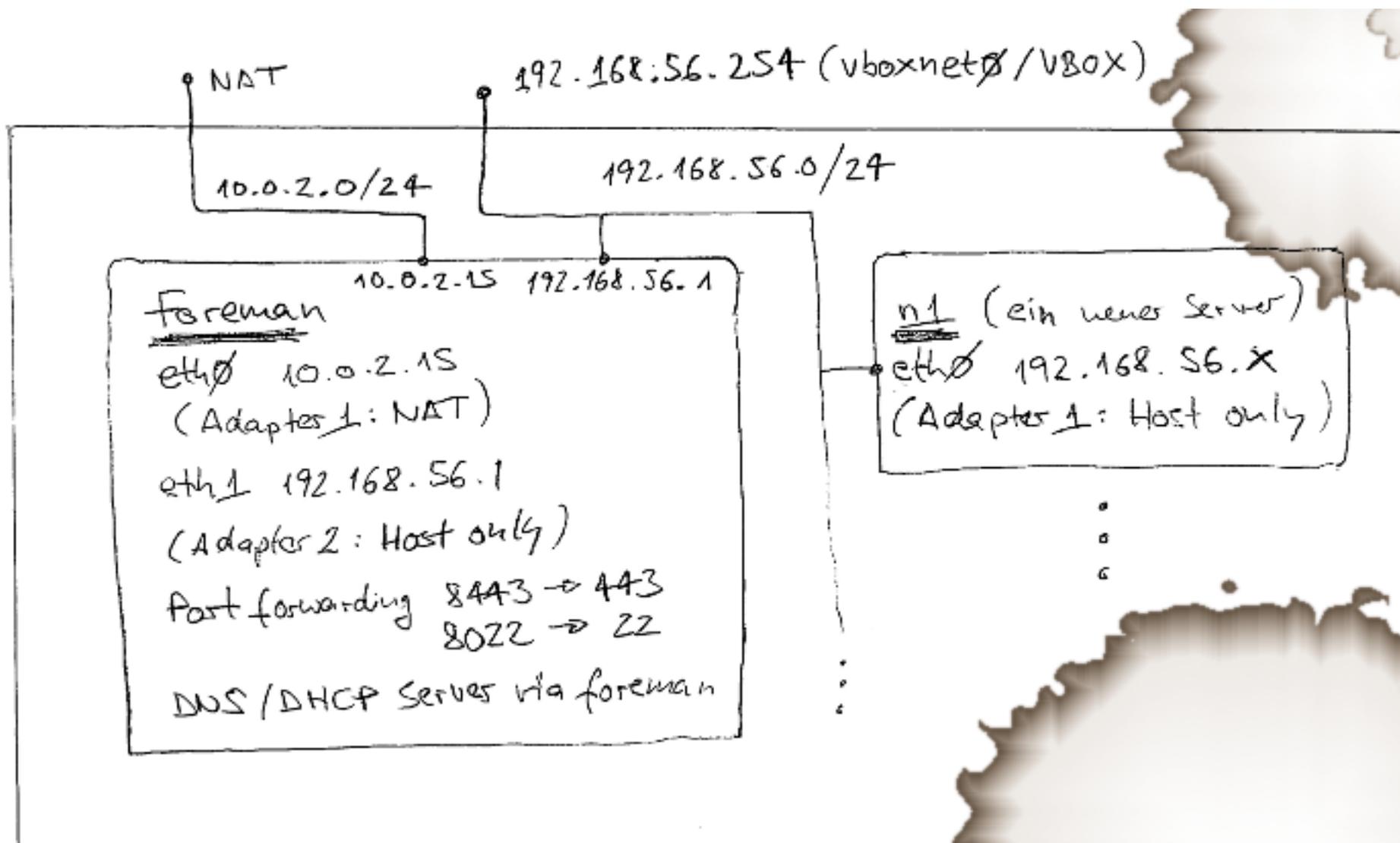
Willkommen bei Foreman

Version 1.5.2

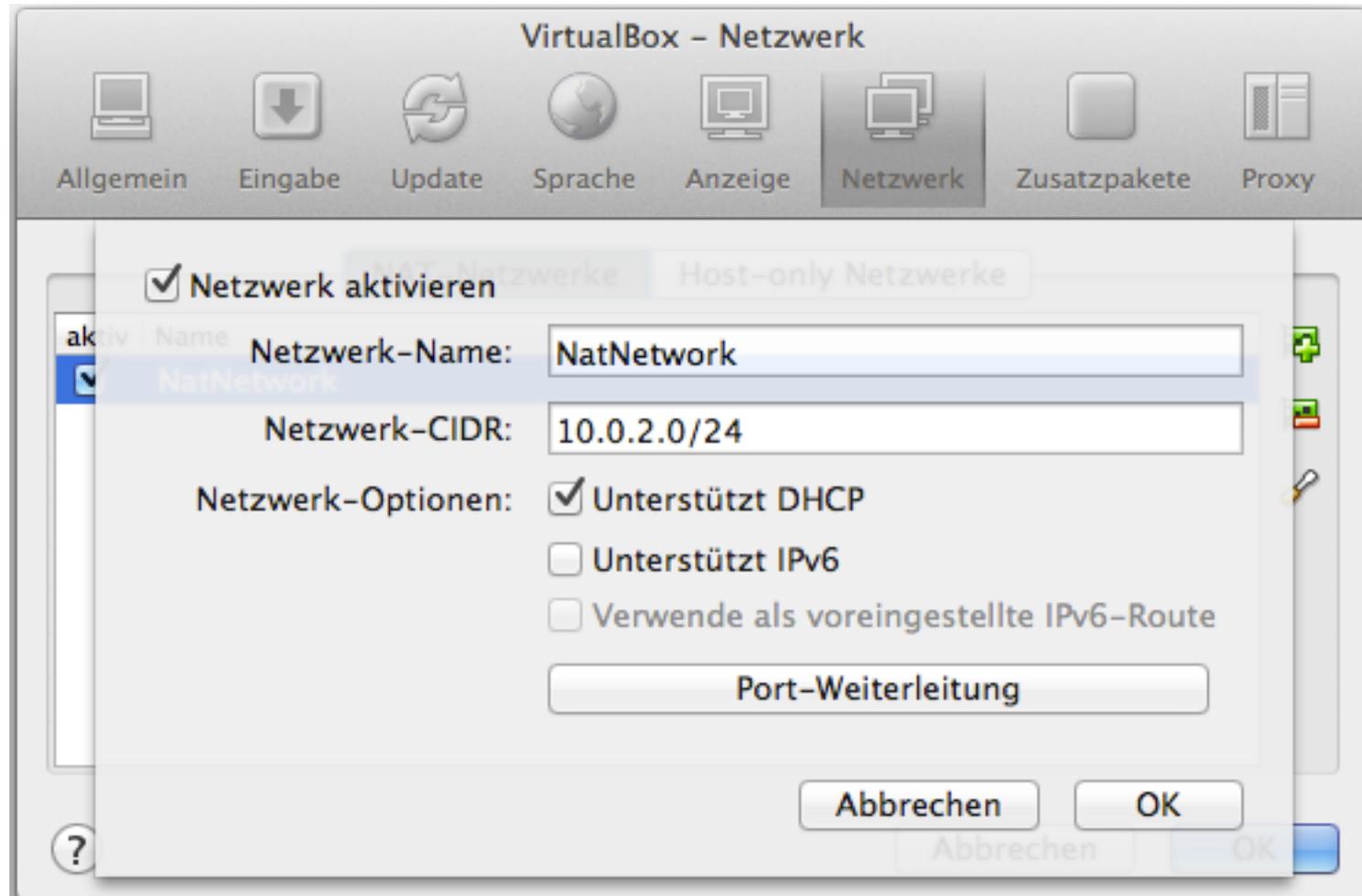
- **VirtualBox, darin:**
 - 1 Foreman-Server
 - N weitere virtuelle Server, die über Foreman/Puppet konfiguriert werden
- **Foreman (<http://www.foreman.org>)**
 - mit TFTP-, DNS- und DHCP-Server konfigurieren
 - mit Puppet
 - unter Ubuntu Server 14 LTS

Die folgende Modell-Stellung dient zur Aufstellung eines Test-Systems und ist in dieser Form nicht(!) für einen Produktionseinsatz ratsam!

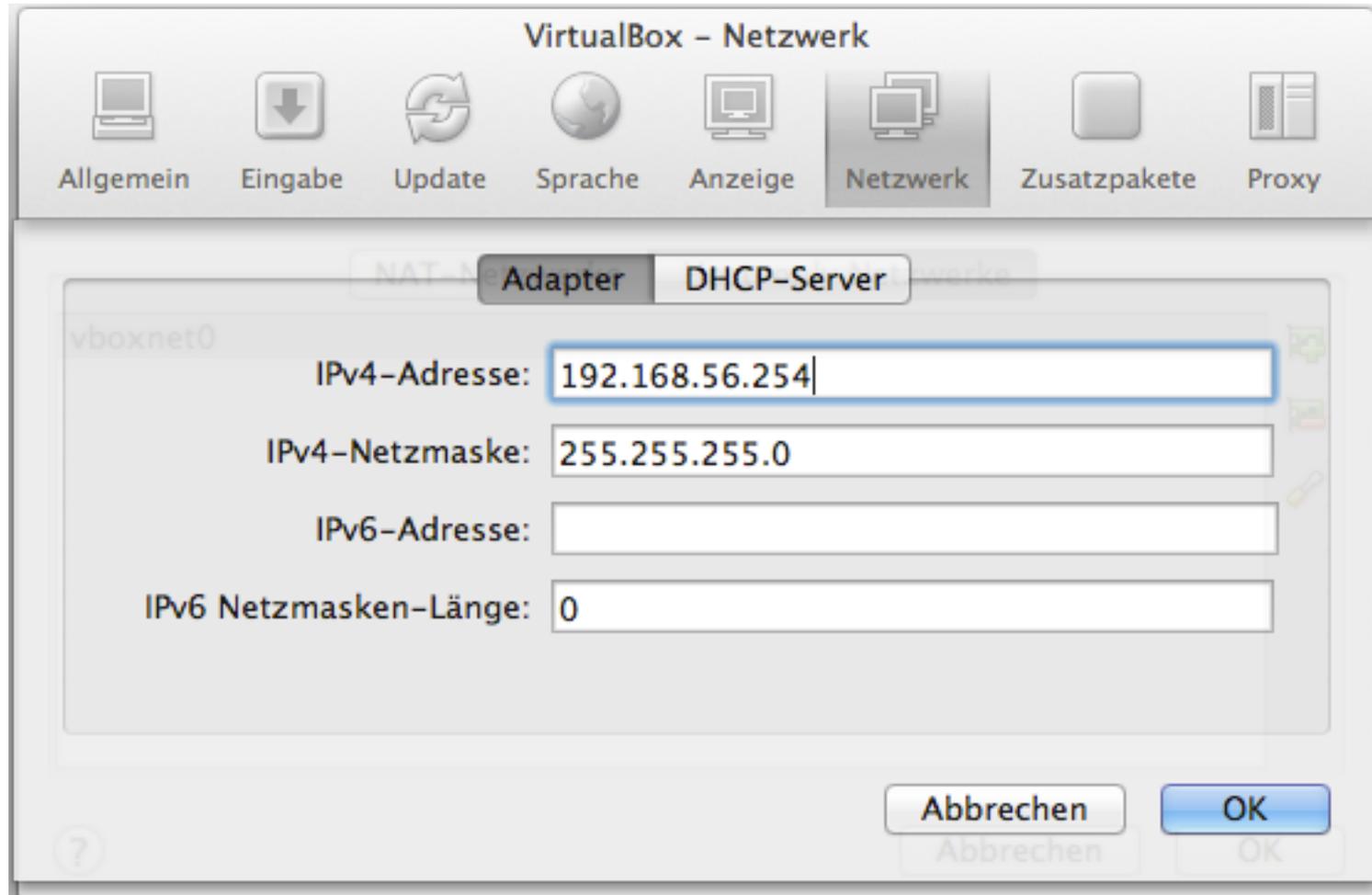
VirtualBox: Planungsskizze



VirtualBox-Konfiguration: NatNetwork



VirtualBox-Konfiguration: Host only Adapter (Adapter)



VirtualBox-Konfiguration: Host only Adapter (DHCP)



VirtualBox: Konfiguration Netzwerk Foreman-Server

Foreman2 - Netzwerk

Allgemein System Anzeige Massenspeicher Audio **Netzwerk** Ports Gemeinsame Ordner

Name	Protokoll	Host-IP	Host-Port	Gast-IP	Gast-Port
SSH	TCP		8022		22
https	TCP		8443		443

Angegeschlossen an: NAT

Name:

Erweitert

Adaptertyp: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous-Modus: erlauben für allen VMs und den Host

MAC-Adresse: 08002786F9D1

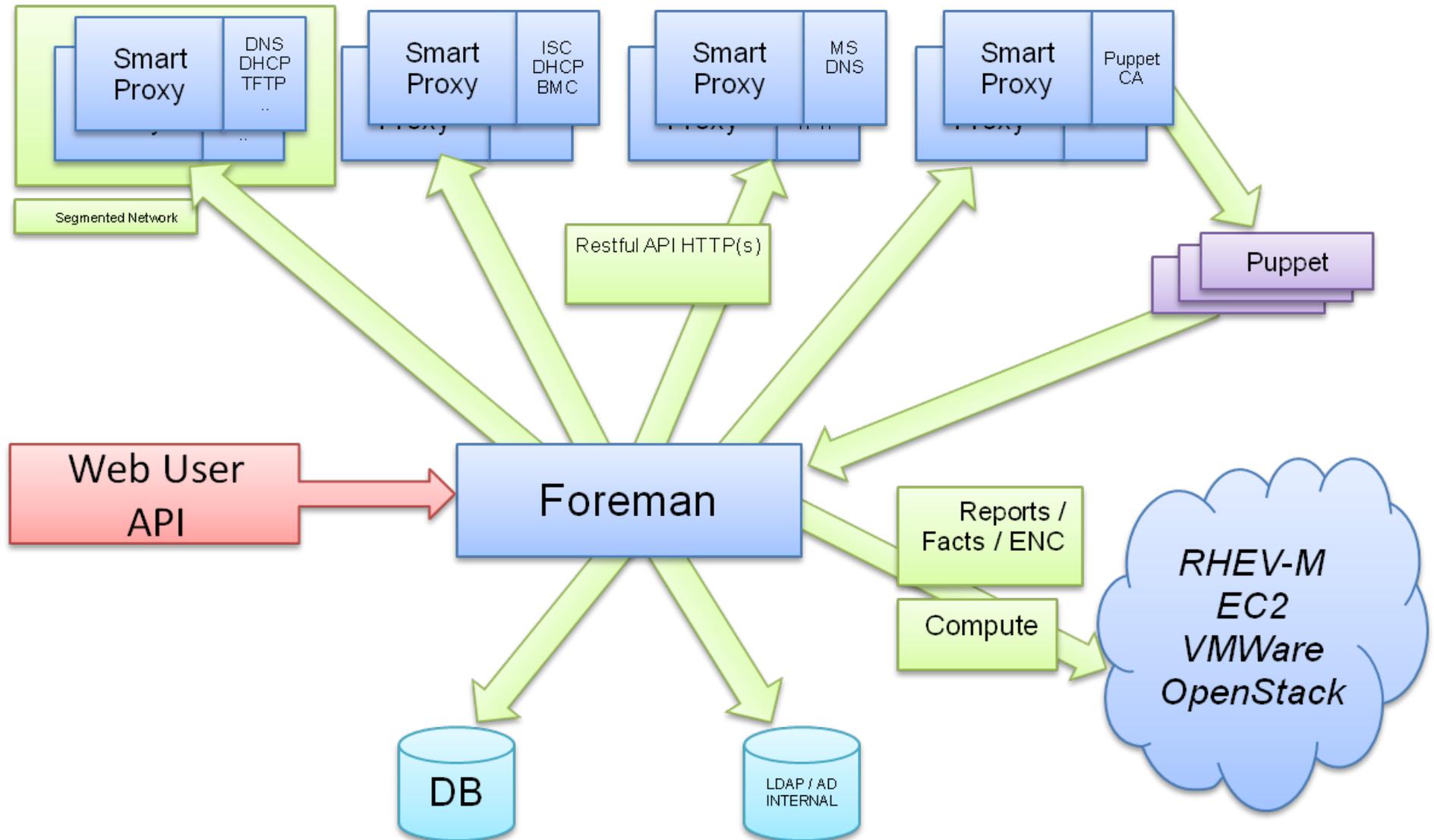
Kabel verbunden

Port-Weiterleitung

Abbrechen OK

? Abbrechen OK

Foreman-Architektur



- **DHCPd**
 - Verteilung von IP-Adressen (und ggf. PXEboot images) an Server
- **TFTPd**
 - Provisionierung von PXEboot images
- **DNS**
 - Eintrag in Domain via DNS rewriting (Sicherheit!?)
- **Back-End**
 - Puppet, Postgres und viel Ruby, ruby, ruby – und rack,...
- **Monitoring / Metriken**
 - Ja, via Web-Interface. Ressourcen-Monitoring!

Pakete installieren (siehe auch: www.foreman.org):

```
echo "deb http://deb.theforeman.org/ trusty 1.5" > /etc/apt/  
sources.list.d/foreman.list && echo "deb http://  
deb.theforeman.org/ plugins 1.5" >> /etc/apt/sources.list.d/  
foreman.list && wget -q http://deb.theforeman.org/pubkey.gpg  
-O- | apt-key add - && apt-get update && apt-get install  
foreman-installer
```

Foreman installer prüft auf

```
hostname -f == hostname `factor fqdn` ggf. hostname anpassen!,
```

Ggf. IP forwarding aktivieren (in Testumgebung ratsam):

```
net.ipv4.ip_forward=1 (/etc/sysctl.conf)
```

```
iptables -t nat -I POSTROUTING -s 192.168.56.0/24 -j  
MASQUERADE (/etc/rc.local)
```

Dann:

```
foreman-installer -i
```

An dieser Stelle die Konfiguration mit der Planung abgleichen (hier u.a.):

- 32. Set tftp, current value: true
- 37. Set tftp_servername, current value: 192.168.56.1
- 38. Set dhcp, current value: true
- 39. Set dhcp_managed, current value: true
- 40. Set dhcp_interface, current value: eth1
- 41. Set dhcp_gateway, current value: 192.168.56.1
- 42. Set dhcp_range, current value: 192.168.56.10 192.168.56.29
- 43. Set dhcp_nameservers, current value: default
- 49. Set dns, current value: true
- 50. Set dns_managed, current value: true
- 51. Set dns_provider, current value: nsupdate
- 52. Set dns_interface, current value: eth1
- 59. Set dns_forwarders, current value: ["<ip_of_dns_server>"]

Allgemein

Name: n12
Betriebssystem: Other Linux (64-bit)

System

Hauptspeicher: 1024 MB
Bootreihenfolge: **Platte, Netzwerk**
Beschleunigung: VT-x/AMD-V, Nested Paging, PAE/NX

Vorschau



Anzeige

Massenspeicher

Controller: IDE
Primärer Master: n12.vdi (normal, 8,00 GB)
Sekundärer Master: [CD/DVD] leer

Audio

Netzwerk

Adapter 1: Intel PRO/1000 MT Desktop **(Host-only Adapter, 'vboxnet0')**

USB

Gemeinsame Ordner

Beschreibung

Foreman – Login

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken



Benutzername

Passwort

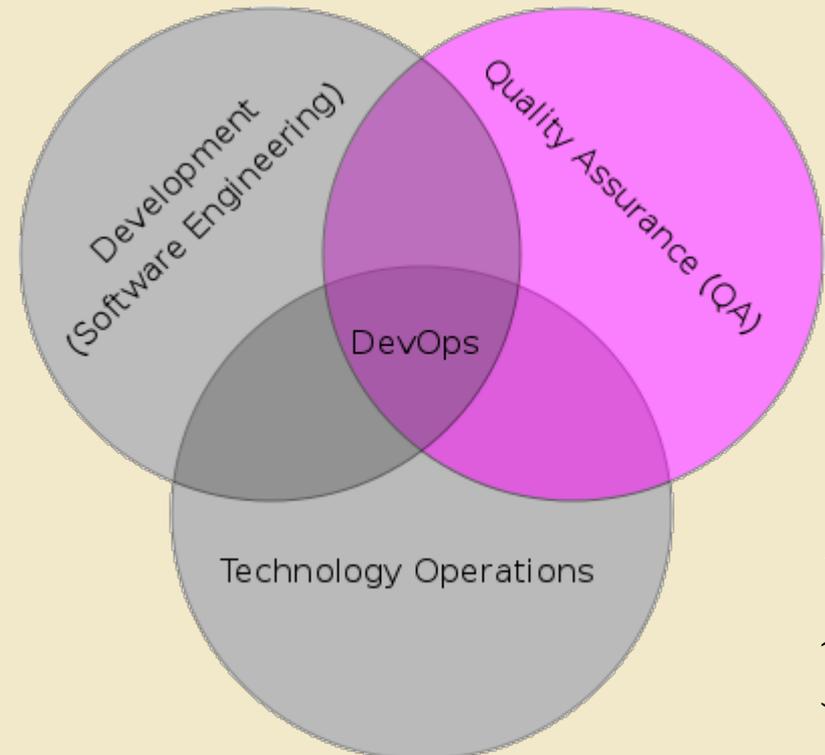
Anmeldung

Willkommen bei Foreman

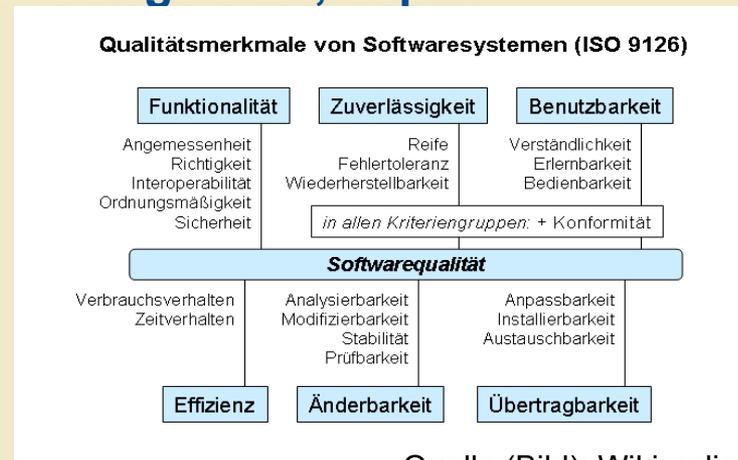
Version 1.5.2

DevOps - Qualitätssicherung ...

- Durch agile (Entwicklungs-)Methoden stehen Releases, Patches und Updates schneller zur Verfügung
- Anforderungen an ein Qualitätsmanagement steigen; stabiler Betrieb muss gewährleistet sein
- **Neue Organisationskultur!**
 - Fehler/Störungen gemeinsam analysieren und beheben
 - Keine gegenseitige Schuldzuweisung



- **Softwarequalität als Begriff wenig greifbar**
- **Normen**
 - IEEE 610.12-1990,
 - ISO 8402,
 - ISO/IEC 9126
 - ...
- **Daraus werden Modelle abgeleitet; bspw.:**



Quelle (Bild): Wikipedia

„Testen ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden. [...] Testen [ist] ein destruktiver, ja geradezu ein sadistischer Prozess [...]“ Glenford Myers

- **Statische Verfahren**
 - (statische) Code-Analyse
- **Dynamische Verfahren**
 - „Tests zur Laufzeit“
 - Black-Box-Verfahren
 - Äquivalenzklasse – Bildung von Testfallgruppen, wobei einzelne Elemente der jeweiligen Gruppe repräsentativ für alle anderen Elemente herangezogen werden können / wenige Testfälle erübrigen die Eingabe vieler weiterer
 - Grenzwertanalyse – Tests durch Betrachtung der Grenzfälle einer Eingabe
 - ...
- **Modultests / Unittests** – frühe Teststufe, Tests einzelner Bereiche
- **Integrationstests** – Tests von einander abhängiger *Units*
- **Systemtests** – Tests des Systems gegen die gesamte Anforderung
- **Abnahmetests / Akzeptanztests** – Tests durch den*die Auftraggeber*in

- **Durch agile Methoden muss häufiger getestet werden**

„Die alte Leier“ ?

- **Wann wird getestet?**
- **Wer testet?**
- **Testfälle entwickeln?**

= Tests sind

- Zeitintensiv
- Kostenintensiv

Agiles Testen

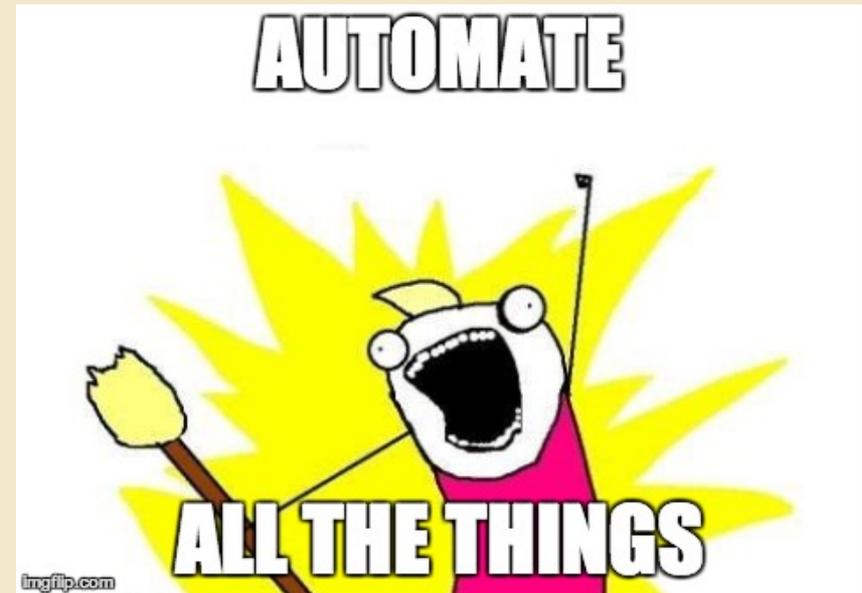
- **Schnelles Feedback / Enge Zusammenarbeit**
- **Logische Konsequenz: Automatisierung von Tests**
 - Regressionstests (dynamische Verfahren)
 - Wiederholung von Testfällen: Prinzip „Grüne-oder-Rote-Ampel“ (JUnit) bzw. Verifikation / Falsifikation durch SOLL-IST-Werte / *Spezifikation*
 - Regressionstest als Teil der Diversifizierenden Tests (Liggesmeyer): Verifikation / Falsifikation durch „vergleichen der Ergebnisse zweier Tests“
 - Entwicklung / Nutzung von Frameworks zur Testunterstützung
 - Management bspw. mit *Scrum*
 - *Hinweis: exploratives / manuelles Testen kann allgemein nicht vollständig ersetzt werden (Abnahmetests)*

- **Wann wird getestet?** → Während der Entwicklung bzw. direkt nach Freigabe
- **Wer testet?** → Entwickler*innen → Nach der Freigabe: Tester*innen
- **Testfälle entwickeln?** → Test-Driven-Development, allgemein im Vorfeld

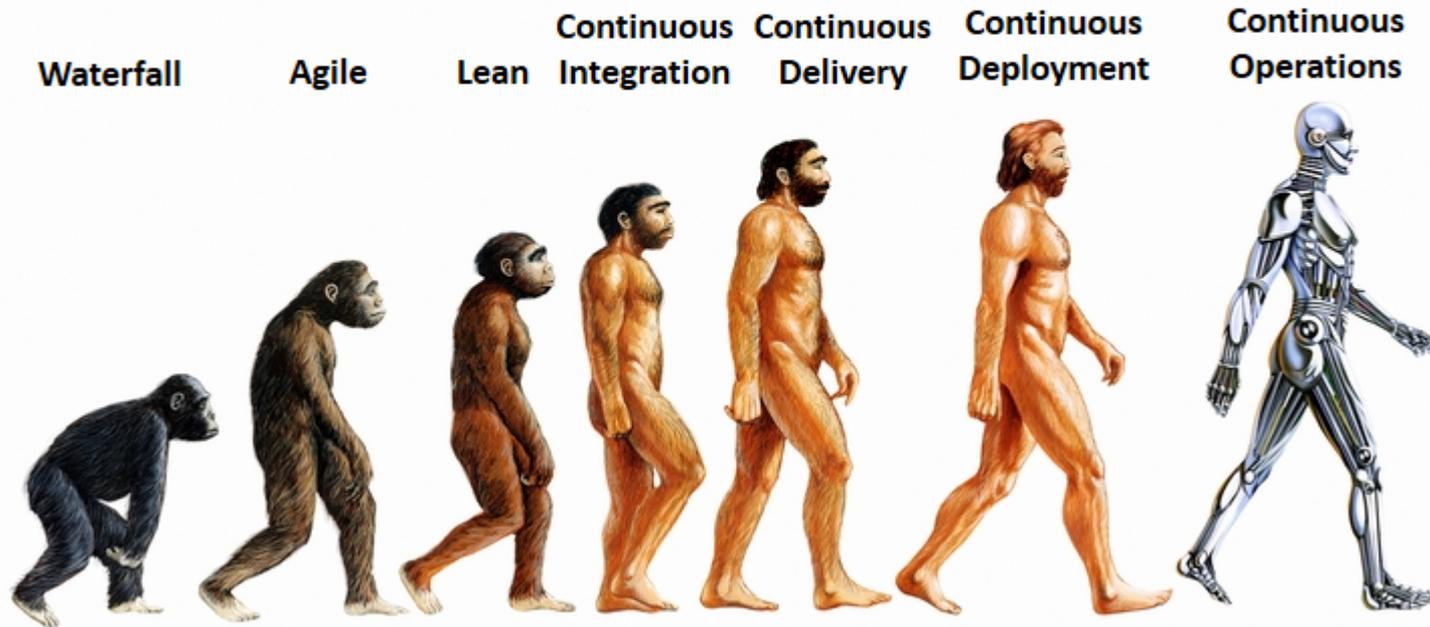
= Tests sind

- Zeitintensiv → Zeitdruck 
- Kosten werden *allerdings* verringert; Fehler können und werden im allgemeinen schneller entdeckt
- Risiko: Sackgassen; denen kann durch Modularisierung entgegen gewirkt werden

- **Tester*innen sind in das Entwicklungsteam integriert**
- **Continuous Integration**
 - Versionsverwaltung
 - Automatische Builds
 - Automatische Tests
 - Automatische Auslieferung / Verteilung der (neuen) Softwarepakete an die Tester*innen / Kunden*innen



DevOps Movement



Quelle (Bild): <http://www.virtualizationpractice.com/continuous-operations-for-zero-downtime-deployments-22680/>

- Entlastung der Beschäftigten
- Automatisierung liefert eine (Software-)Metrik; Informationen über die Stabilität der *Builds* sind greifbarer
 - Anforderungen erfüllt?
 - Fehler behoben?
 - Haben Änderungen am Code Effekte auf andere Bestandteile der Software?
 - ...

Automatisierte Softwaretests – „Die 7 schlechtesten Ideen für Testautomatisierung“

- **Erfolg nach wenigen Sprints erwarten** – Testautomatisierung nicht als Feature verkaufen, sondern in den Entwicklungskosten mit führen
- **Testwerkzeugen blind vertrauen** – Mitarbeiter*innen müssen geschult werden; Grenzen des Tools berücksichtigt?; Anwendbar auf den gesamten Betrieb?
- **Schreiben der Testskripte als Nebenbeschäftigung ansehen**
- **Testdaten irgendwo in Testfällen vergraben**
- **Testautomatisierung nur mit Benutzeroberflächen in Verbindung bringen**
- **Soll-Ist-Vergleich unterschätzen** – Tests mit gleichem Datenzustand-/bestand wiederholbar machen; Ergebnisparser nur einsetzen, wenn es nicht anders geht
- **(Un-)Testbarkeit der Applikation einfach hinnehmen** – Design for Testability; Bspw.: Bei der Entwicklung von Weboberflächen eindeutige IDs verwenden

Quelle: [Baumgartner...]

... mit Selenium

„Selenium automates browsers. That's it! [...] [Selenium] is for automating web applications for testing purposes, but is certainly not limited to just that.” Thoughtworks

- **Entwicklergruppe Thoughtworks bzw. Jason Huggins, Sion Stewart, ... (seit 2004)**
 - **Open-Source, Apache 2.0. lizenziert**
 - **Selenium IDE (FireFox Plugin)**
 - ***Selenium Remote Control – deprecated***
 - **Selenium WebDriver**
- 
- Basierend auf
JavaScript**
- **Basis**
 - *JUnit*

Find Elements By:

- ID
- Name
- Link

- Xpath
- CSS

Tools

- Firebug
- Firefinder
- IE Developer Tools
- Google Chrome Developer Tools

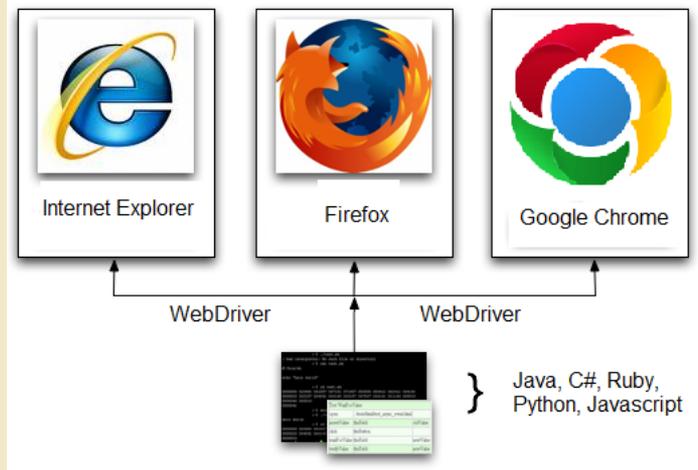
Firefox Plugins

- **Für die Entwicklung: HTML-Tags der Webapplikationen sollten mit IDs versehen sein**
- **Xpath**
 - Für komplexe Abfragen; können Tests verlangsamen
 - Xpath / CSS: Für Dynamische Bezeichner bzw. wenn keine IDs vorhanden sind

Hands-on mit Selenium IDE & WebDriver

- **Standard: <http://www.w3.org/TR/2013/WD-webdriver-20130117/>**

„(...) a platform and language-neutral interface that allows programs or scripts to introspect into, and control the behaviour of, a web browser”



- **Design Pattern: *Page Object*** „Jede (zu testende) Webseite wird durch ein *Object* repräsentiert.“
 - Das Page-Object ist das *Userinterface*
 - Funktionen der Webseite werden als Funktionen / Methoden des Page Objects implementiert

- **QA – Framework (HIS)**
 - Implementiert Design-Pattern *Page Object*
 - Wird zur Qualitätssicherung seitens der HIS eingesetzt
 - Regressionstests
 - Aktuell wird der Einsatz an den Hochschulen erprobt ...

- **Performance / Statistik Anfragezeiten**
- **Updates / Patches**
- **Lokale Anpassung / Nebeneffekte; nächtliche Überprüfung**

Fazit

„[...] [Selenium] is for automating web applications for testing purposes, but is certainly not limited to just that.”

„Boring web-based administration tasks can (and should!) also be automated as well.”

Ein mögliches Vorgehensmodell

Foreman, puppet



git



Selenium



Betriebssystem
Middleware
Framework
Tool
Paket



Versionierte
Anwendung



Versionierte
Anwendungs-
Konfiguration



Versionierte
SAF

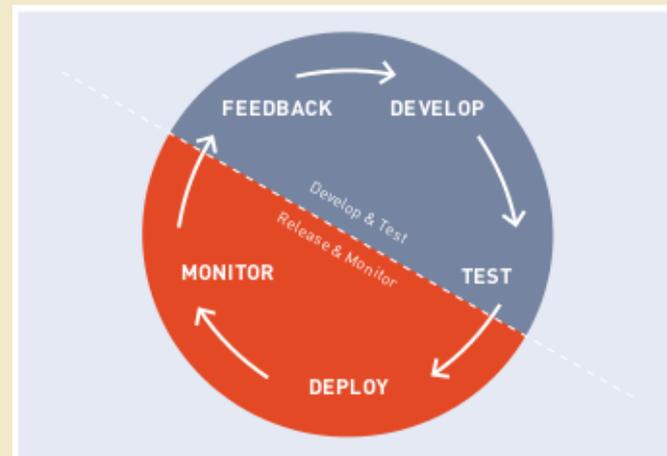


IaaS/PaaS

SaaS

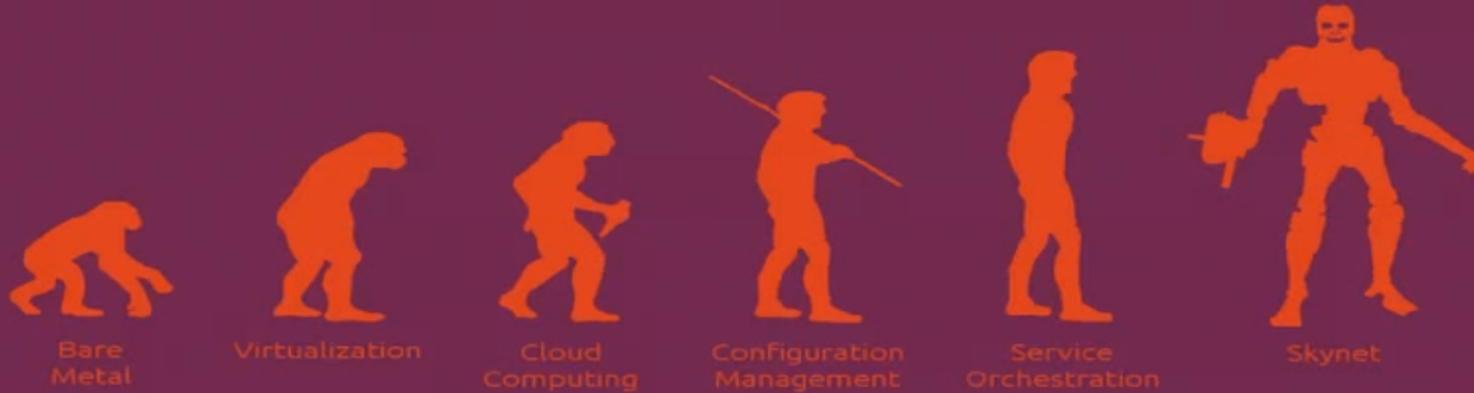
Qualitäts-
sicherung

- **Auslagern von (System-)Konfigurationen in Versionskontrolle; Erfahrung aus der Softwaretechnik nutzen**
- **Automatisierung von Systemtests**
- **Proaktiver Austausch von Informationen**
 - Nicht *Developing* und *Operating* beiden Personengruppen gleichermaßen auferlegen, sondern Transparenz zwischen den Beteiligten verbessern



Quelle (Bild): Materna Monitor, Ausgabe 4/2014

DevOpsolution



juju.ubuntu.com
DevOps Distilled™

Quelle (Bild): <http://octopress.pztrick.com/blog/2013/07/28/a-year-of-development/>

Vielen Dank!

- **Selenium**

- [Thoughtworks] <http://www.seleniumhq.org> (englisch)

- **Literatur**

- [Baumgartner...] Baumgartner, Klöckner, Pichler, Seidl, Tanczos, Agile Testing, Carl Hansen Verlag München, 2013
- [LinuxMagazin01] <http://www.linux-magazin.de/Ausgaben/2014/09/Testgetrieben/%28language%29/ger-DE>
- [LinuxMagazin02] <http://www.linux-magazin.de/Ausgaben/2014/09/Das-Prinzip/%28language%29/ger-DE>
- [LinuxMagazin03] <http://www.linux-magazin.de/Ausgaben/2005/04/Aufmerksamer-Beobachter>
- [MaternaMonitor] Materna Monitor, Ausgabe 4/2014: http://www.materna.de/DE/Service/Newsroom/Monitor/monitor_node.html
- [Myers] G. J. Myers, Methodisches Testen von Programmen, 7. Auflage, Oldenbourg Verlag München Wien, 2001
- [Schneider] Kurt Schneider, Abenteuer Software Qualität, 2. Auflage, dpunkt.verlag GmbH, 2012
- [Walls] Mandi Walls, „Building a DevOps Culture“: <http://chimera.labs.oreilly.com/books/1234000000810>
- <http://www.installationpage.com/selenium/how-to-run-selenium-headless-firefox-in-ubuntu/> -- Selenium Headless
- <http://theagileadmin.com/what-is-devops/>
- https://github.com/DevOpsBootcamp/website/blob/master/source/11_devops.rst