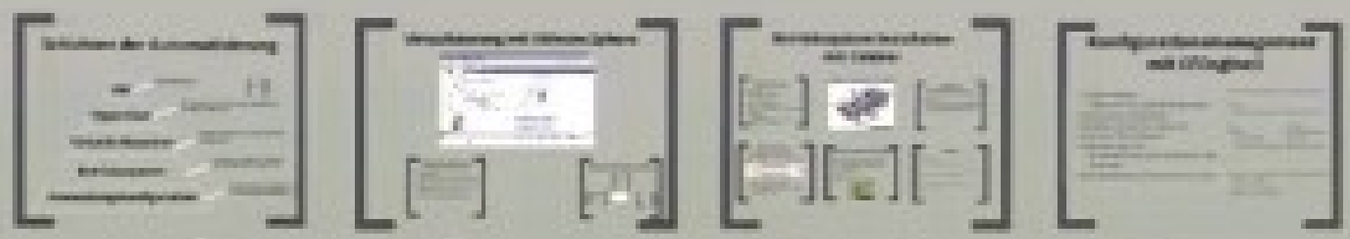
TNG TECHNOLOGY CONSULTING

Infrastructure as Code

Virtualisierung und Automatisierung - wie man sich ein Rechenzentrum programmiert

Fazit

Werkzeuge

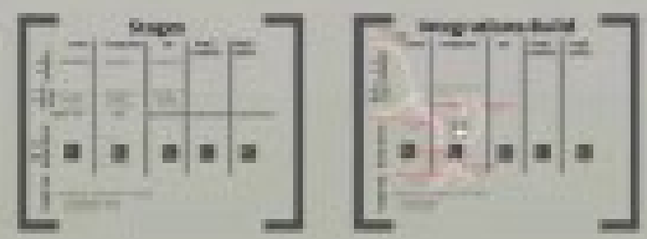


Ressourcen Planung, Virtualisierung

PXE Boot, DHCP Netzwerkinstallation

Service + Stage Datei Konfiguration

Continuous Delivery



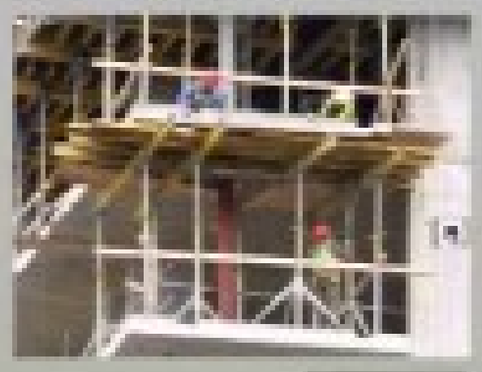
Über uns

Mathias Münch

Jörg Meltzer

Infrastructure as Code

Infrastructure as a Service ⇄ Infrastructure as Code



vmware cobbler CFEngine®

Infrastructure as Code

Infrastructure as a Service

am schönen Wolkenhimmel

Infrastructure as Code

[ ]

http://nikhewitt.blogspot.de/

Infrastructure as Code

CFEngine®

vmware®

- VMware vSphere 4.1
- VMware vSphere SDK for Perl


cobber

cobber

CFEngine®

```
vmcreate.pl --vmname "testVM" --mac0
00:50:56:7e:45:1b --tpl L
```

```
cobbler system add
--name=<x10110.domain.de>
--profile=<RHEL6>
--mac=<00:50:56:7e:45:1b>
--ip=<172.22.113.50>
--hostname=<x10110.domain.de>
```

```
bundle agent resolver
{
files:
  "/etc/resolv.conf"
    copy_from => remote(
"${global.cfmaster}","$(global.filestore)/resolver/resolv.co
nf","false"),
    classes => if_ok("resolv_conf_ok"),
    perms => mog("644", "root", "root");

reports:
  resolv_conf_ok::
    "Name resolution configured correctly.";
}
```

Infrastructure as Code

- VMware vSphere 4.1
- VMware vSphere SDK for Perl

Ops machen Code

Devs machen Infrastruktur

"Infrastructure as code". I love the phrase. Where devops is a word that is sadly open to so much (mis)interpretation, "Infrastructure as code" is pretty clear. Treat your infrastructure as code. Programmable. Testable. Deployable. (John E. Vincent)

Ops machen Code

- von Devs lernen
  - Agile Methoden
  - Continuous Delivery
  - Automatisierte Tests

Devs machen Infrastruktur

- von Ops lernen
  - Packaging
  - Konfigurationsmanagement
  - Monitoring

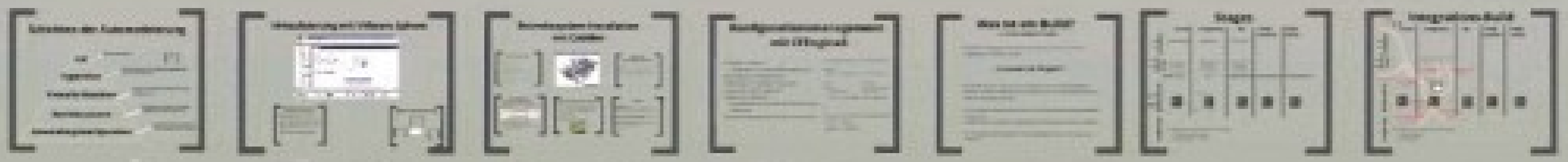
TNG TECHNOLOGY CONSULTING

Infrastructure as Code

Virtualisierung und Automatisierung - wie man sich ein Rechenzentrum programmiert

Fazit

Werkzeuge



Ressourcen Planung, Virtualisierung

PXE Boot, DHCP Netzwerkinstallation

Service + Stage Datei Konfiguration

Continuous Delivery

Schichten der Automatisierung

HW

Blade, SAN, Switch

Hypervisor

Betriebssystem der Virtualisierungsplatform
z.B. VMware ESXi, Xen

Virtuelle Maschine

erstellen, zerstören, virtuelle Hardware konfigurieren

Betriebssystem

Installserver: Kickstart, PXE Boot
Dienste: DNS, DHCP,LDAP, VCS

Anwendungskonfiguration

Konfigurationsmanagement-Tools: Puppet/Chef/CFEngine

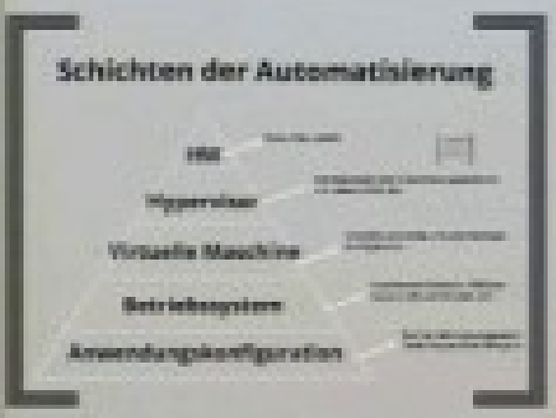
Clonen und patchen oder neu bauen?

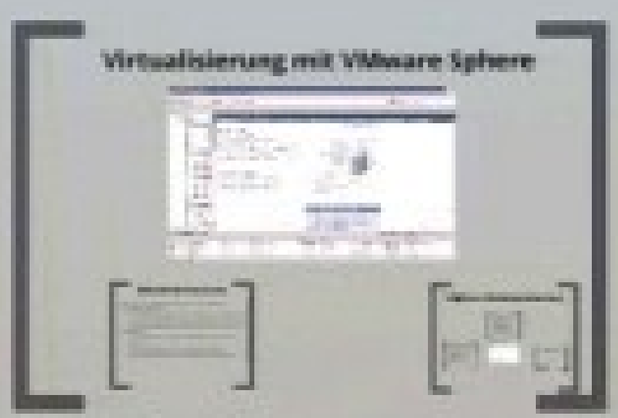
Patchen schon damals eine Bad Practice

"'Real Programmers' are reluctant to actually edit a program that is close to working. They find it much easier to just patch the binary object code directly"
(Ed Post, Real Programmers Don't Use PASCAL)

ein Rechenzentrum

Werkzeuge



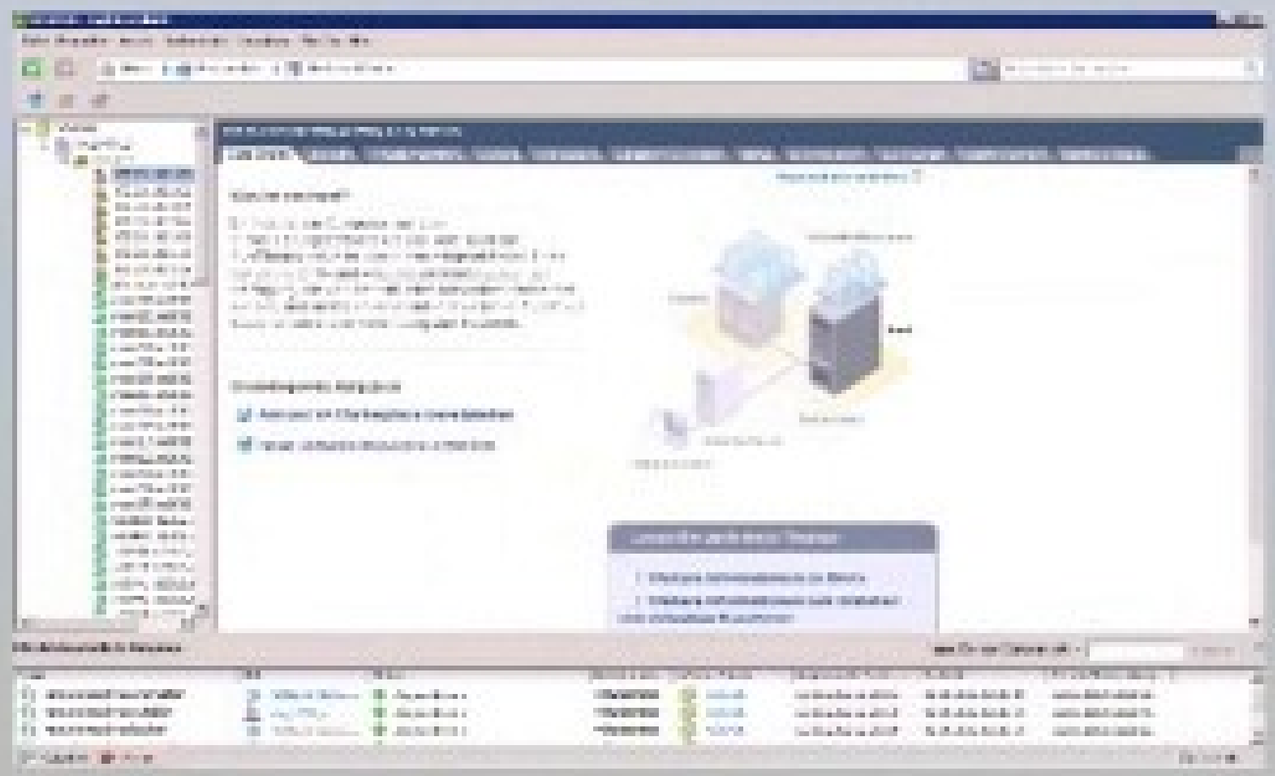


Ressourcen Planung, Virtualisierung

PXE Boot, DHCP Netzwerkinstallation

Service + Stage Datei Konfiguration

Virtualisierung mit VMware Sphere



Bekannte Features

- Mehrere OS Installationen parrele auf einen Rechner
- Hardware wird virtuell
  - Virtualisierung ermöglicht die Verwaltung von Hardware als Pools (CPU, Speicher, Storage und Netzwerkressourcen
  - Virtuelle Maschinen sind flexibel, solange der Pools reichen
- Robustheit
  - Virtuelle Maschinen migrieren bei Ausfall auf andere Rechner
  - Bei Ausfällen starten VMs auf anderen Rechnern
  - Konfiguration dafür so leicht wie möglich und flexibel

vSphere Automatisierung

Bekannte Features

- Mehrere OS Installationen passen auf einen Rechner.
- Hardware wird teilbar:
  - Virtualisierung ermöglicht die Betrachtung der Hardware als Pool aus CPU-, Speicher-, Storage- und Netzwerkressourcen.
  - Virtuelle Maschinen sind baubar, solange die Pools reichen.
- Robustheit:
  - Virtuelle Maschinen migrieren bei Überlast auf andere Rechner.
  - Bei Ausfällen starten VMs auf anderen Rechnern.
  - Konfiguration dafür ist leicht einstellbar und änderbar.

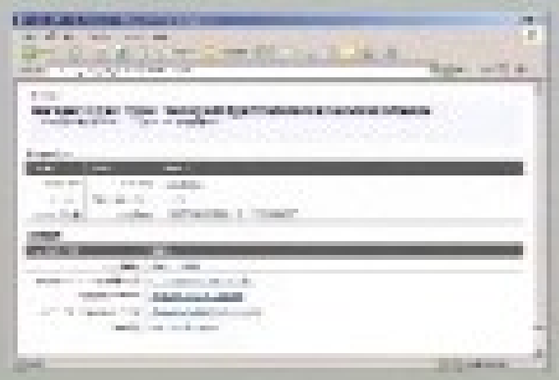
vSphere Automatisierung

vSphere Perl-SDK

- sehr mächtiger Wrapper um vCenter Soap-Api
- Schnelle Entwicklung
  - Beispielscripte nutzen
  - gute Online-API-Documentation
  - API bietet auch synchrone Operationen
  - VMware Community Scripte
- schlecht
  - Codequalität Beispielscripte
  - POD Documentation in Modulen fehlt
  - keine VMware Shell analog zur PowerShell CLI

Virtualisierung erleichtert Automatisierung

- Zusätzliche Maschinen lassen sich ohne Hardwarekauf erstellen.
- Das VM-Deployment wird programmierbar.
- Die Virtualisierungsplattform abstrahiert vom Typ der Hardware. VMs sehen nur einfache "Standard" Hardware.



Beispiel - Reporting

Virtualisierung erleichtert Automatisierung

- Zusätzliche Maschinen lassen sich ohne Hardwarekauf erstellen.
- Das VM-Deployment wird programmiert.
- Die Virtualisierungsplatform abstrahiert von Spezialhardware, VMs sehen nur einfache "Standard" Hardware.

vSphere Perl-SDK

- leichtgewichtiger Wrapper um vCenter Soap Api
- Schnelle Entwicklung
  - Beispielskripte nutzen
    vm{create,destroy,info,control}.pl
  - gute Online API Dokumentation
  - API bietet auch synchrone Operationen
  - VMware Community Skripte
- schlecht
  - Codequalität Beispielskripte
  - POD Dokumentation in Modulen fehlt
  - keine VMware Shell analog zur PowerShell CLI

Managed Object Browser - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address https://<my_esx_server>/vmware.com/mob Go Links

Home

Managed Object Type: ManagedObjectReference:ServiceInstance

Managed Object ID: **ServiceInstance**

Properties

NAME	TYPE	VALUE
capability	Capability	capability
content	ServiceContent	content
serverClock	dateTime	"2007-11-26T21:49:47.291812Z"

Methods

RETURN TYPE	NAME
dateTime	CurrentTime
HostVMotionCompatibility[]	QueryVMotionCompatibility
ServiceContent	RetrieveServiceContent
ProductComponentInfo[]	RetrieveProductComponents
Event[]	ValidateMigration

Done Local intranet

Managed Object Browser - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address https://<my_esx_server>/vmware.com/mob Go Links

Home

Managed Object Type: ManagedObjectReference:ServiceInstance

Managed Object ID: **ServiceInstance**

Properties

NAME	TYPE	VALUE
capability	Capability	capability
content	ServiceContent	content
serverClock	dateTime	"2007-11-26T21:49:47.291812Z"

Methods

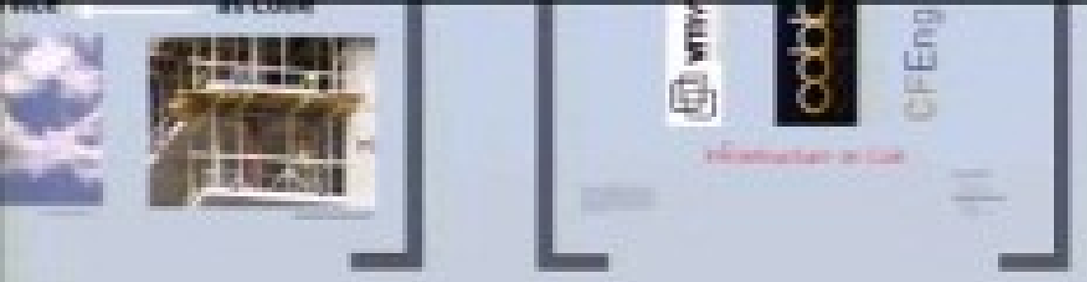
RETURN TYPE	NAME
dateTime	CurrentTime
HostVMotionCompatibility[]	QueryVMotionCompatibility
ServiceContent	RetrieveServiceContent
ProductComponentInfo[]	RetrieveProductComponents
Event[]	ValidateMigration

Done Local intranet

Beispiel - Reporting

```
#!/usr/bin/perl
use strict;
use warnings;

# Step 1: Import the vSphere SDK for Perl Modules.
use VMware::VIRuntime;

# Step 2: (Optional) Define Script-Specific Command-
Line Options.
my %opts = (
    entity => {
    type => "=s",
    help => "ManagedEntity type: VirtualMachine, etc",
    required => 1,
    },
);

Opts::add_options(%opts);
Opts::parse();
Opts::validate();

# Step 3: Connect to the Server.
Util::connect();

# Step 4: Obtain View Objects of Server-Side Managed Objects.
# Obtain all inventory objects of the specified type

my $entity_type = Opts::get_option('entity');
my $entity_views = Vim::find_entity_views(
    view_type => $entity_type);

# Step 5: Process Views and Report Results.
# Process the findings and output to the console
foreach my $entity_view (@$entity_views) {
  my $entity_name = $entity_view->name;
  Util::trace(0, "Found $entity_type:    $entity_name\n");
}

# Step 6: Close the Server Connection.
# Disconnect from the server
Util::disconnect();
```

```
$ report_names.pl --entity VirtualMachine
```



ein Rechenzentrum

Werkzeuge

Ressourcen Planung, Virtualisierung

PXE Boot, DHCP Netzwerkinstallation

Service + Stage Datei Konfiguration

Betriebssystem Installation mit Cobbler

Cobbler Features

- zentraler Installationsserver
- Open Source
- automatisiert OS Installation
  - RHEL, SuSE, Debian
- vereinfachte Handhabung von
  - PXE Boot, TFTP Server, DHCP, DNS, ...
- Flexible Kickstarts mit Snippets und Templating
- Web interface
- Verwaltung
  - Distribution, Profil, System, YUM Repos
- Python, XML-RPC Schnittstelle

Netzwerkinstallation mit Cobbler

Übergang zum Konfigurationsmanagement

- Vorsicht Produktiv, nur das Notigste in Kickstart installieren
- System mit Service und Stage Information impfen
- Letzter Task der OS Installation ist der initiale Lauf des Konfigurationsmanagement Tools

Herkömmliche Netzwerkinstallation

Install Server neuer Server

- Netzwerkadresse via DHCP beziehen
- TFTP download Kernel + Initrd
- Download automatisches Kickstart Bootskript
- Installation durchführen

manuelle Netzwerkinstallation

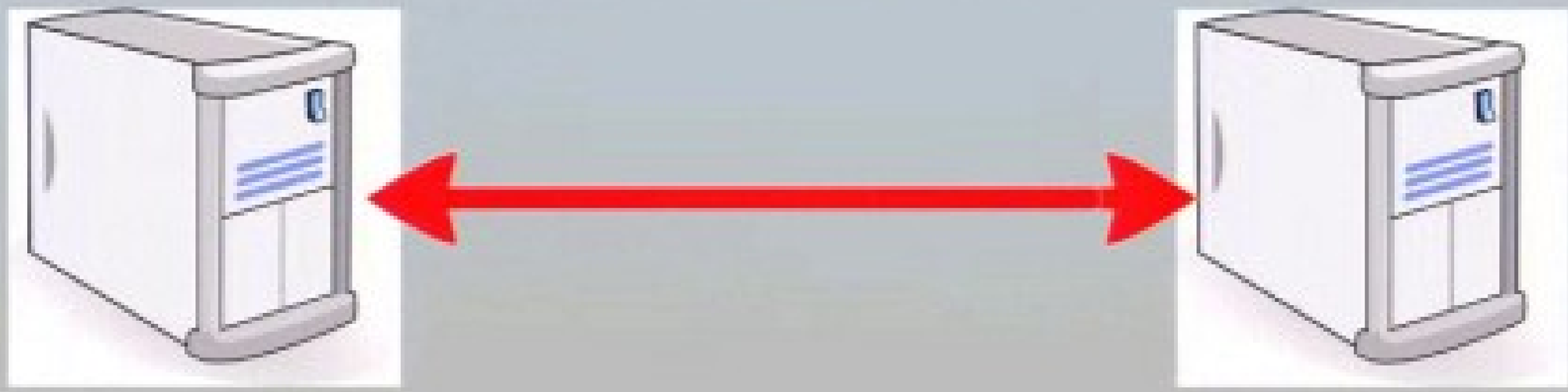
- neue Mac Adresse in /etc/dhcpd.conf eintragen
- PXE Boot Datei auf TFTP Server anlegen
- Copy & Paste Kickstart Datei von einem Template
- DHCPD neustarten
- neuen Server einschalten
- warten
- über Fehler in obiger Procedur ärgern

Beispiel

Herkömmliche Netzwerkinstallation

Install Server

neuer Server



- Netzwerkaddresse via DHCP beziehen
- TFTP download Kernel + Initrd
- Download automatisches Kickstart Bootskript
- Installation durchführen

manuelle Netzwerkinstallation

- neue Mac Addresse in /etc/dhcpd.conf eintragen
- PXE Boot Datei auf TFTP Server anlegen
- Copy & Paste Kickstart Datei von einem Template
- DHCPD neustarten
- neuen Server einschalten
- warten
- über Fehler in obiger Prozedur ärgern

Netzwerkinstallation mit Cobbler

http://en.wikipedia.org/wiki/File:The_giant_wenger.jpg

Cobbler Features

- zentraler Installationsserver
- Open Source
- automatisiert OS Installation
  - RHEL, SuSE, Debian
- vereinfacht Handhabung von
  - PXE Boot, TFTP Server, DHCP, DNS, ...
- Flexible Kickstarts mit Snippets und Templating
- Web Interface
- Orchestrierung
  - Distribution, Profil, System, YUM Repos
- Python, XML-RPC Schnittstelle

Beispiel

```
# import mounted dvd
cobbler import --path=/mnt --name=CentOS-6 --kickstart=/var/lib/cobbler/kickstarts/default.ks
cobbler distro report --name CentOS-6

cobbler profile add --name latest --comment "profile for development hosts"
cobbler repo    add --name latest --mirror /var/www/yum/latest --comment "yum repo for development"
...

cobbler profile add --name int --comment "profile for integration hosts"
cobbler repo    add --name int --mirror /var/www/yum/int --comment "yum repo for integration"
cobbler system  add --name int1 --profile int --interface eth0 --mac ... --ip ...


cobbler profile add --name prod --comment "profile for prod1 hosts"
cobbler repo    add --name prod --mirror /var/www/yum/prod --comment "yum repo for prod"
cobbler system add  --name prod1 --profile prod

# dhcp, tftp... synchronisieren
cobbler sync
```

Übergang zum Konfigurationsmanagement

- Vorsicht Featuritis, nur das Nötigste in Kickstart installieren
- System mit Service und Stage Information impfen
- Letzter Task der OS Installation ist der initiale Lauf des Konfigurationsmanagement Tools

Konfigurationsmanagement mit CFEngine3

- CFEngine Policies
  - "Makefile" für Systemkonfiguration
- konvergente Konfiguration
- Policyhub ist dummer Fileserver
- autonome Client Agenten
- deklarative Sprache
  - beschreibt, was gemacht werden soll, nicht wie
- Modularisierung mit Bundles und Bodys

```
bunde agent shadow {    # wie Subroutine

classes:
  "solinux" expression => "linux|solaris";

files:                          # Phase
solinux::                       # Einschränkung
  "/etc/shadow"                 # promise
    perms => p_shadow;          # promise body

}

body perms p_shadow { # snippet, template
  mode => "640";
  owners => {"root"};
  groups => {"shadow"};
}
```

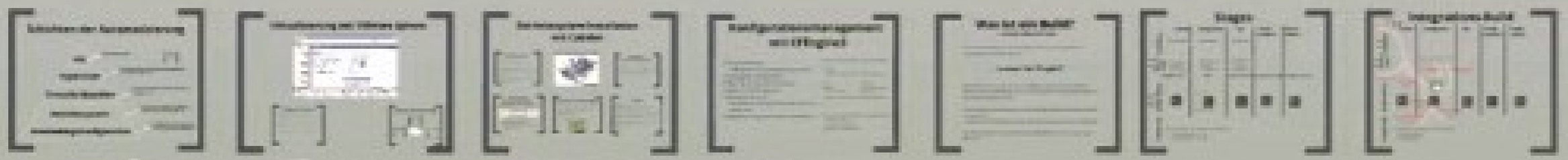
TNG TECHNOLOGY CONSULTING

Virtualisierung und Automatisierung - wie man sich ein Rechenzentrum programmiert

Infrastructure as Code

Fazit

Werkzeuge



Ressourcen Planung, Virtualisierung

PXE Boot, DHCP Netzwerkinstallation

Service + Stage Datei Konfiguration

Continuous Delivery

Was ist ein Build?

... in einem Software Projekt

- Ein Build ist Sequenz von Schritten, die aus Source Code ein deploybares Artefakt erstellt.

... in einem IaC Projekt?

- Ein Build ist eine Sequenz von Schritten, die für eine bestimmte Paketzusammenstellung und Sammlung von Konfigurationspolicies virtuelle Maschinen baut.

- Der CFEngine Masterserver checkt CFEngine Policies in bestimmter Version aus.
- Cobbler baut YUM Repos mit bestimmter Paketzusammenstellung.

- Ein Skript löst die komplette Neuinstallation für eine Gruppe von VMs aus.

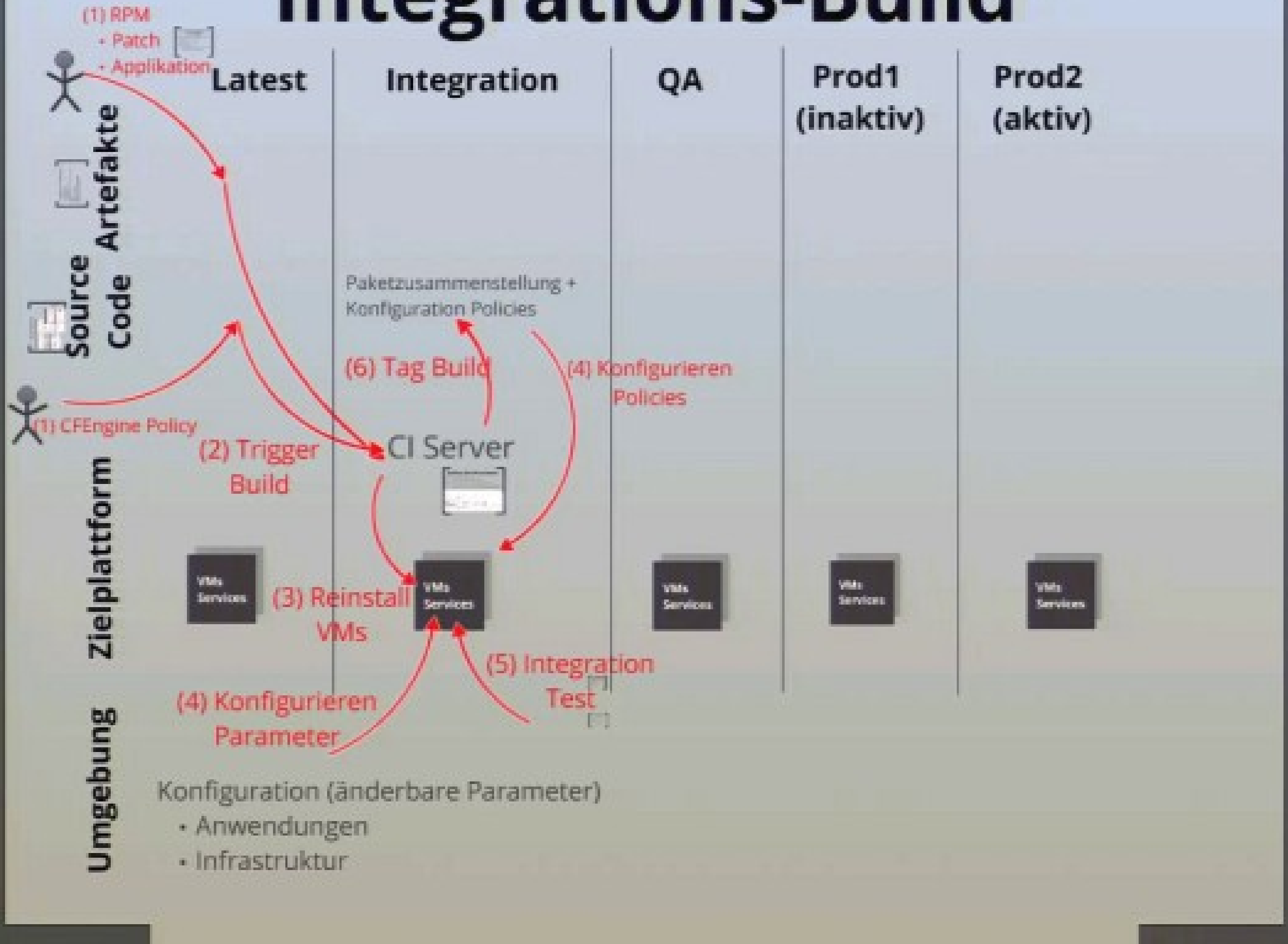
Stages

	Latest	Integration	QA	Prod1 (inaktiv)	Prod2 (aktiv)
Artefakte	yum/latest	yum/int	yum/qa		
Source Code	packages.csv CFEngine masterfiles	packages.csv CFEngine masterfiles	packages.csv CFEngine masterfiles	...	...
Version	trunk	r4711	Integration-Build-42	Integration-Build-42	Integration-Build-41
Zielplattform	VMs Services	App-VMs VMs Services Infrastruktur-VMs	VMs Services	VMs Services	VMs Services
Umgebung					

Konfiguration (änderbare Parameter)

- Anwendungen
- Infrastruktur

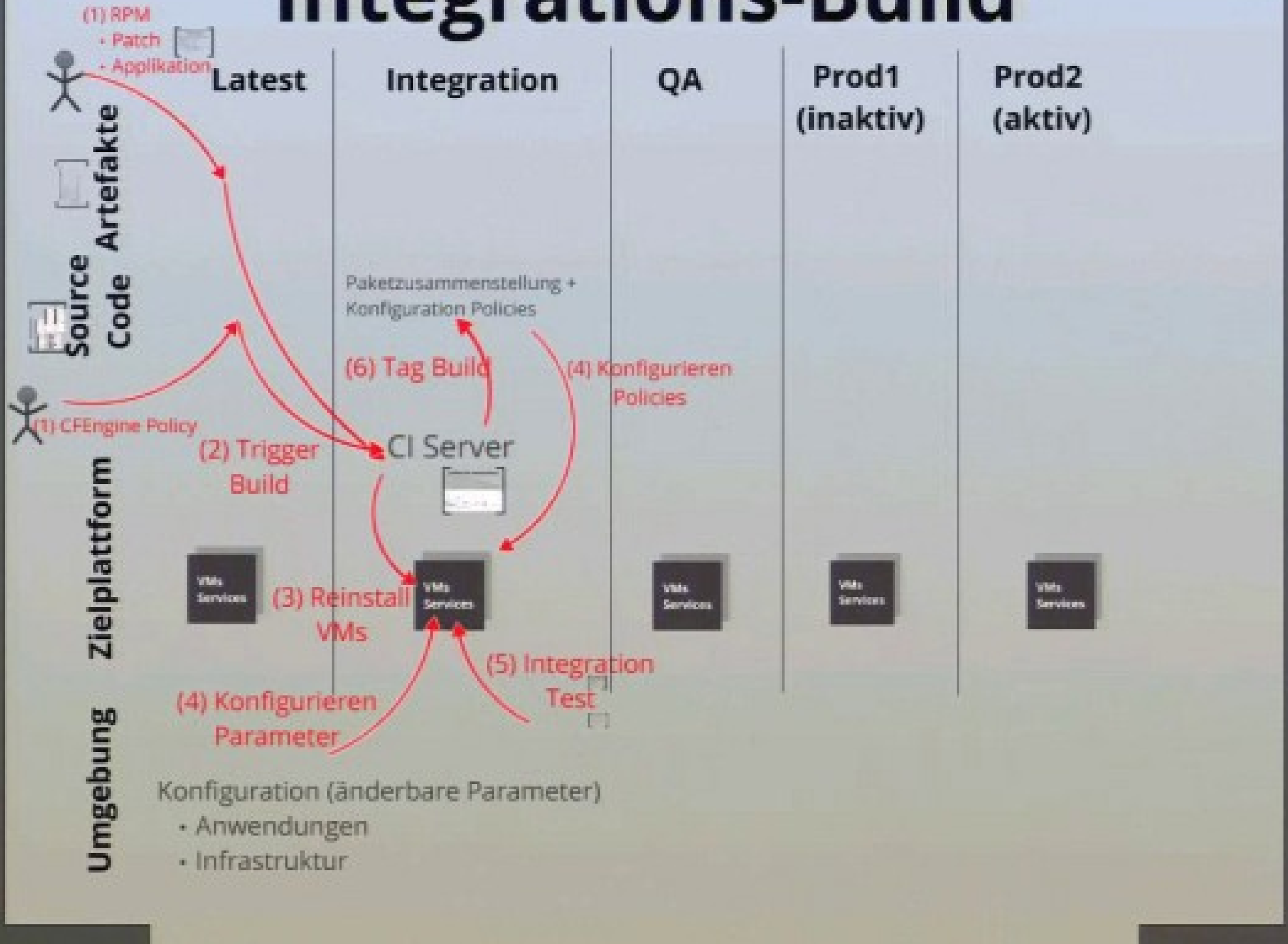
Integrations-Build



Artefakt Versionierung und Paket Auswahl

- Typischer Artefaktname <Name>-<Major>.<Minor>.<Release>
- Strategie bei der Artefaktauswahl für Entwicklungsumgebung
  - immer die höchste Version installieren
  - aktuell bleiben
  - Konfigurationsaufwand minimieren
- Problem bei Bugfixes
  - Lösungsidee Feature Toggling
    - Bugfixes und Features werden gleichzeitig ausgeliefert.
    - Neue Features sind jedoch über die Umgebungskonfiguration initial deaktiviert.

Integrations-Build

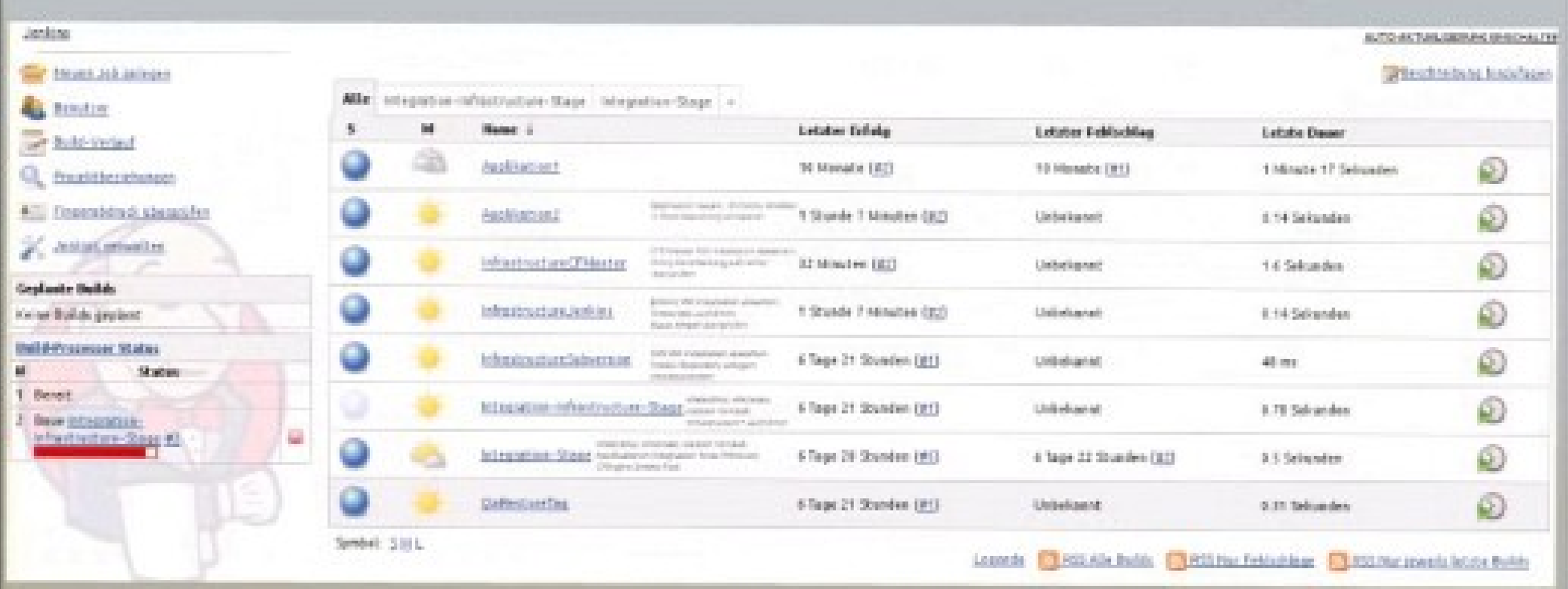


RZ Integration und Projekt Integration

- Software Projekte benötigen eigenen Integrationsbuild
- Projekt Integration
  - baut Applikations RPM
  - testet neue Features
- RZ Integration
  - testet die in Produktion aktivierten Features

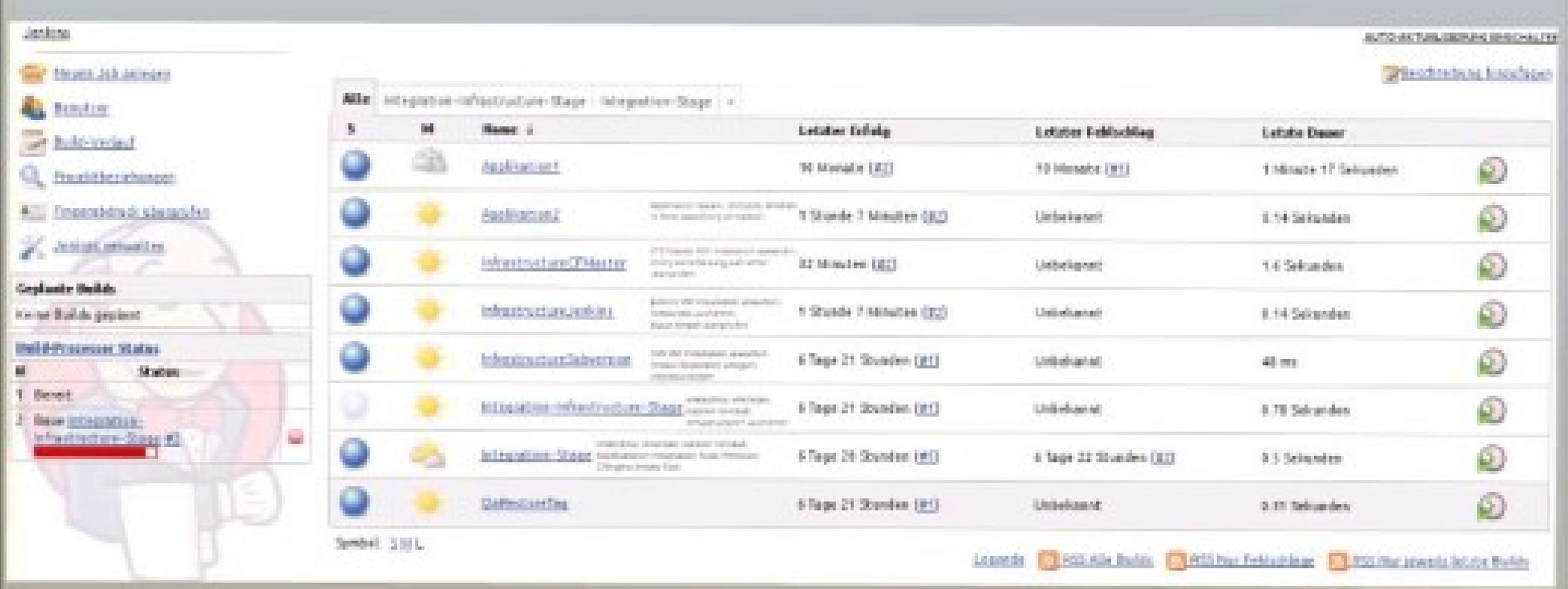
Jenkins für Infrastruktur

- CI Server bauen Softwareprojekte
  - ... und jetzt auch virtuelle Maschinen
- Jenkins Jobs reinstallieren virtuelle Maschinen und
  - testen, ob die Infrastruktur Maschinen noch baubar sind
  - führen funktionale Tests für fachlichen Anwendungen durch



Jenkins für Infrastruktur

- CI Server bauen Softwareprojekte
  - ... und jetzt auch virtuelle Maschinen
- Jenkins Jobs reinstallieren virtuelle Maschinen und
  - testen, ob die Infrastruktur Maschinen noch baubar sind
  - führen funktionale Tests für fachlichen Anwendungen durch



	Applikation2	Applikation bauen, Unittests, Artefakt in Yum Repository einspielen
	InfrastructureCFMaster	CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen
	InfrastructureJenkins	Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen
	InfrastructureSubversion	SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen
	Integration-Infrastructure-Stage	vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen
	Integration-Stage	vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnesse), CFEngine Smoke Test

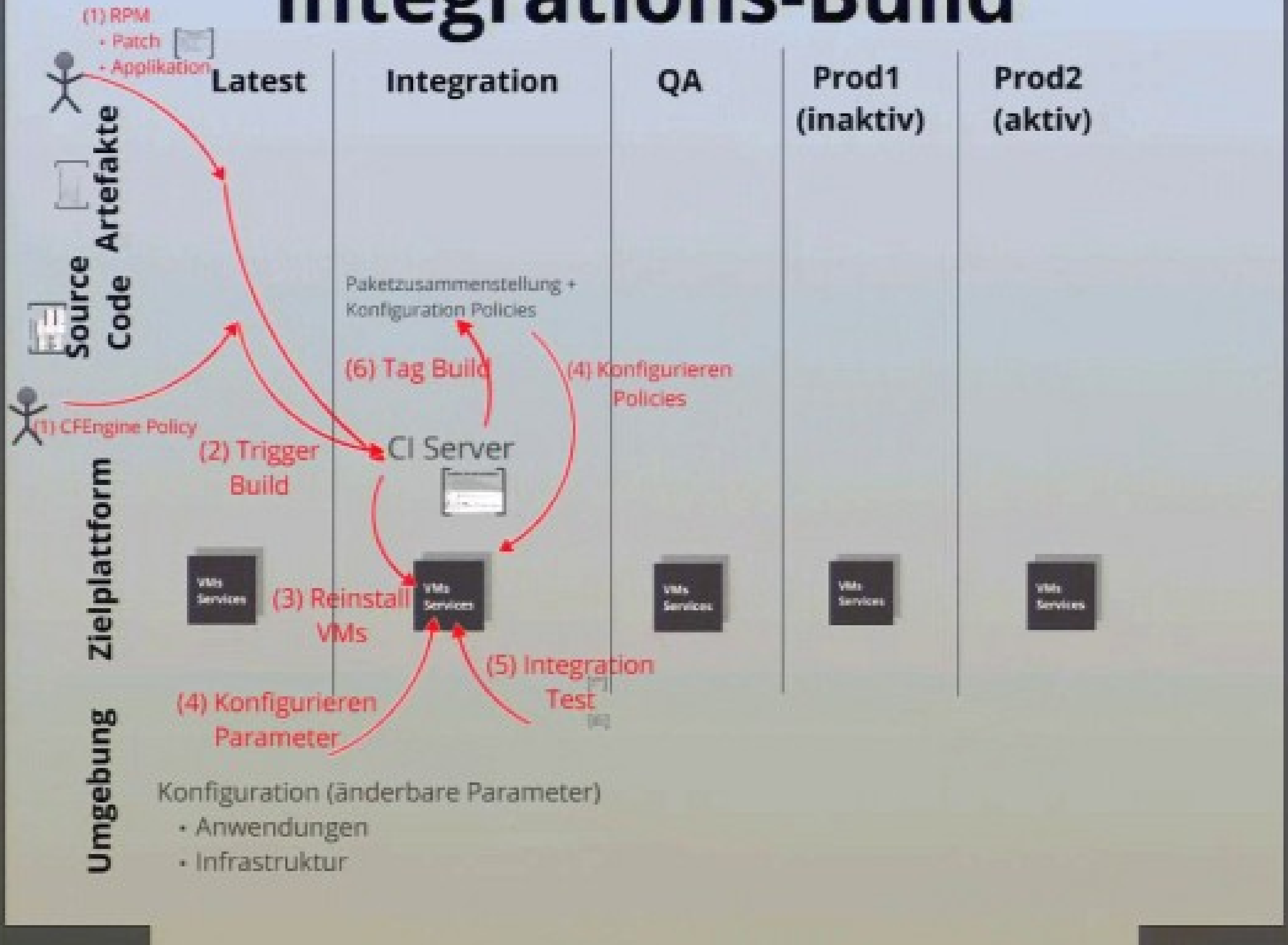
	Applikation2	Applikation bauen, Unittests, Artefakt in Yum Repository einspielen
	InfrastructureCFMaster	CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen
	InfrastructureJenkins	Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen
	InfrastructureSubversion	SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen
	Integration-Infrastructure-Stage	vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen
	Integration-Stage	vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnesse), CFEngine Smoke Test

	Applikation2	Applikation bauen, Unittests, Artefakt in Yum Repository einspielen
	InfrastructureCFMaster	CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen
	InfrastructureJenkins	Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen
	InfrastructureSubversion	SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen
	Integration-Infrastructure-Stage	vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen
	Integration-Stage	vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnesse), CFEngine Smoke Test

	Applikation2	Applikation bauen, Unittests, Artefakt in Yum Repository einspielen
	InfrastructureCFMaster	CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen
	InfrastructureJenkins	Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen
	InfrastructureSubversion	SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen
	Integration-Infrastructure-Stage	vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen
	Integration-Stage	vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnesse), CFEngine Smoke Test

	Applikation2	Applikation bauen, Unittests, Artefakt in Yum Repository einspielen
	InfrastructureCFMaster	CFE Master VM Installation abwarten, Policy Verarbeitung auf Fehler überprüfen
	InfrastructureJenkins	Jenkins VM Installation abwarten, Smoke Job ausführen, blaue Ampel überprüfen
	InfrastructureSubversion	SVN VM Installation abwarten, Smoke Repository anlegen, checkout testen
	Integration-Infrastructure-Stage	vmdestroy, vmcreate, cobbler reinstall, Infrastructure* ausführen
	Integration-Stage	vmdestroy, vmcreate, cobbler reinstall, Applikationen Integration Tests (Fitnesse), CFEngine Smoke Test

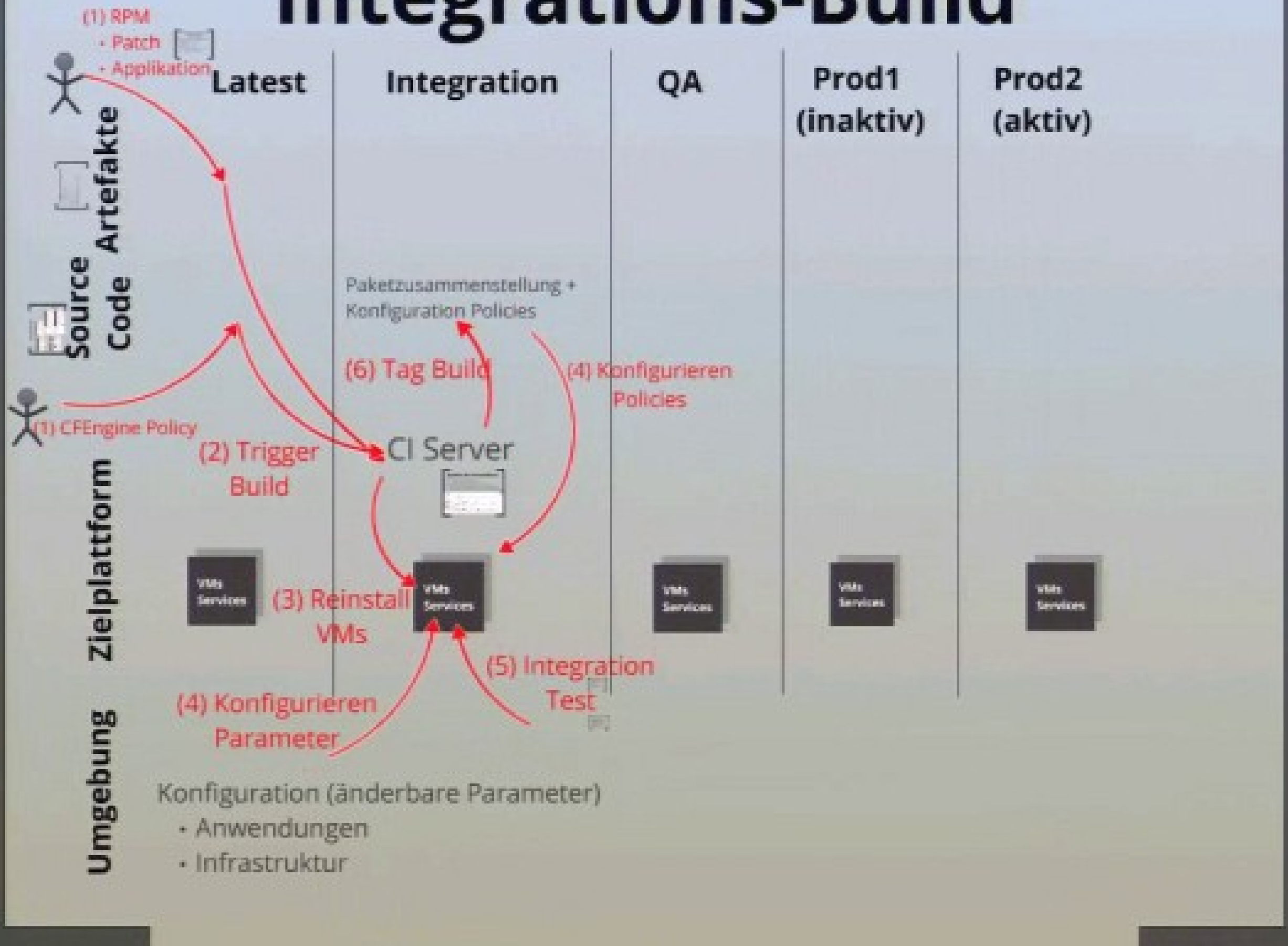
Integrations-Build



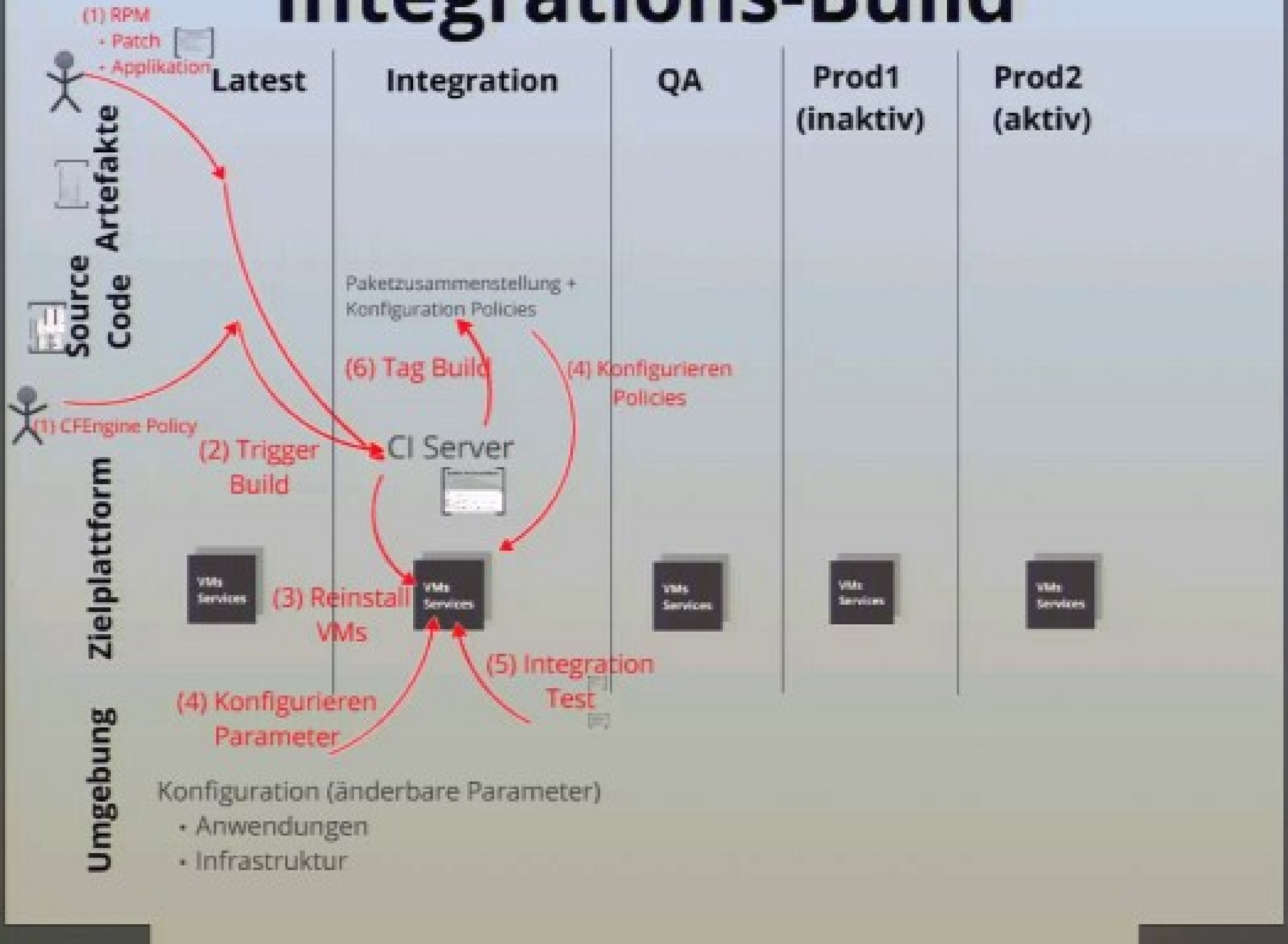
Hartes Staging

- Produktionsinstallation ohne Stress
- Standardisierte Umgebungen
  - in Entwicklungsumgebung Flexibilität auf Anforderung
    - CFEngine Daemon deaktivieren
    - ggf. zusätzliche YUM Repositories nutzen
- Anwendungen müssen Feature Toggling unterstützen
- Oberste Priorität, schnelle Pipeline, häufig stabile Integrationsumgebung
  - möglicher Konflikt
    - Sicherheitspatches täglich in Produktion bringen
    - Anwendung oder CFEngine Policy debuggen

Integrations-Build



Integrations-Build



Das Ganze gibt es wirklich

Infrastructure as Code Projekt

- Kunde: Internetdienstleister, IT Abteilung ~100 MA
- vorher
  - Umgebungen unterschiedlich
  - manuell installiert
  - Hardware schlecht ausgelastet
  - lange Deployment Zyklen
- Aufgabe
  - Rechenzentrum von Grund auf neu bauen
- Team
  - 2 x TNG, 2 x Sysadmins, Tester, Anwendungsentwickler, Netzwerker
- Zeit: 1 Jahr

Akzeptanz der IaC Umgebung

- Errungenschaften
  - Automatisierung mit klarem Schichtenmodell
  - VMs bauen in 5-7 Minuten
  - großes Vertrauen in Umgebungen
  - tägliche Deployments in Produktion
- Schwierigkeiten
  - Prozesse kontinuierlich überprüfen und ändern
  - Herkömmliches CFEngine gibt wenig Struktur vor
    - Zeit in eigene High Level CFEngine Bibliothek und Dev Umgebung investieren

Das Ganze gibt es wirklich

Infrastructure as Code Projekt

- Kunde: Internetdienstleister, IT Abteilung ~100 MA
- vorher
  - Umgebungen unterschiedlich
  - manuell installiert
  - Hardware schlecht ausgelastet
  - lange Deployment Zyklen
- Aufgabe
  - Rechenzentrum von Grund auf neu bauen
- Team
  - 2 x TNG, 2 x Sysadmins, Tester, Anwendungsentwickler, Netzwerker
- Zeit: 1 Jahr

Akzeptanz der IaC Umgebung

- Errungenschaften
  - Automatisierung mit klarem Schichtenmodell
  - VMs bauen in 5-7 Minuten
  - großes Vertrauen in Umgebungen
  - tägliche Deployments in Produktion
- Schwierigkeiten
  - Prozesse kontinuierlich überprüfen und ändern
  - Herkömmliches CFEngine gibt wenig Struktur vor
    - Zeit in eigene High Level CFEngine Bibliothek und Dev Umgebung investieren

Infrastructure as Code Projekt

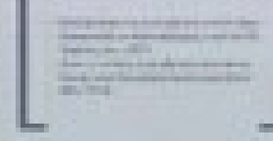
- Kunde: Internetdienstleister, IT Abteilung ~100 MA
- vorher
  - Umgebungen unterschiedlich
  - manuell installiert
  - Hardware schlecht ausgelastet
  - lange Deployment Zyklen
- Aufgabe
  - Rechenzentrum von Grund auf neu bauen
- Team
  - 2 x TNG, 2 x Sysadmins, Tester, Anwendungsentwickler, Netzwerker
- Zeit: 1 Jahr

Akzeptanz der IaC Umgebung

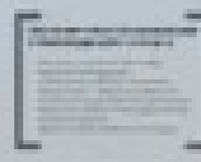
- Errungenschaften
  - Automatisierung mit klarem Schichtenmodell
  - VMs bauen in 5-7 Minuten
  - großes Vertrauen in Umgebungen
  - tägliche Deployments in Produktion

- Schwierigkeiten
  - Prozesse kontinuierlich überprüfen und ändern
  - Herkömmliches CFEngine gibt wenig Struktur vor
    - Zeit in eigene High Level CFEngine Bibliothek und Dev Umgebung investieren

Devops is About CAMS



Culture



Automation

Measurement

Sharing

John Willis
Damon Edwards

Also werden Infrastrukturprojekte jetzt Entwicklungsprojekte - nicht ganz.

- Der Code ist nahe an der wahren Welt.
- Es gibt viele Abhängigkeiten.
- Schreib mal schnell einen Stub für einen Netzwerk-Switch - das geht so einfach nicht.
- Trotz Virtualisierung bleiben viele gemeinsame Ressourcen - Testen ist schwierig wie bei Code mit vielen Singletons.
- Speziell Unit-Tests sind nicht wirklich gelöst.

Tests vs. Monitore - Wo ist der Unterschied?

Tests:

- stellen einen definierten Zustand her,
- sind irgendwann beendet,
- suchen nach Fehlern aufgrund von bekannten, gewollten Veränderungen,
- laufen in spezieller Test- oder CI-Umgebung.

Monitore:

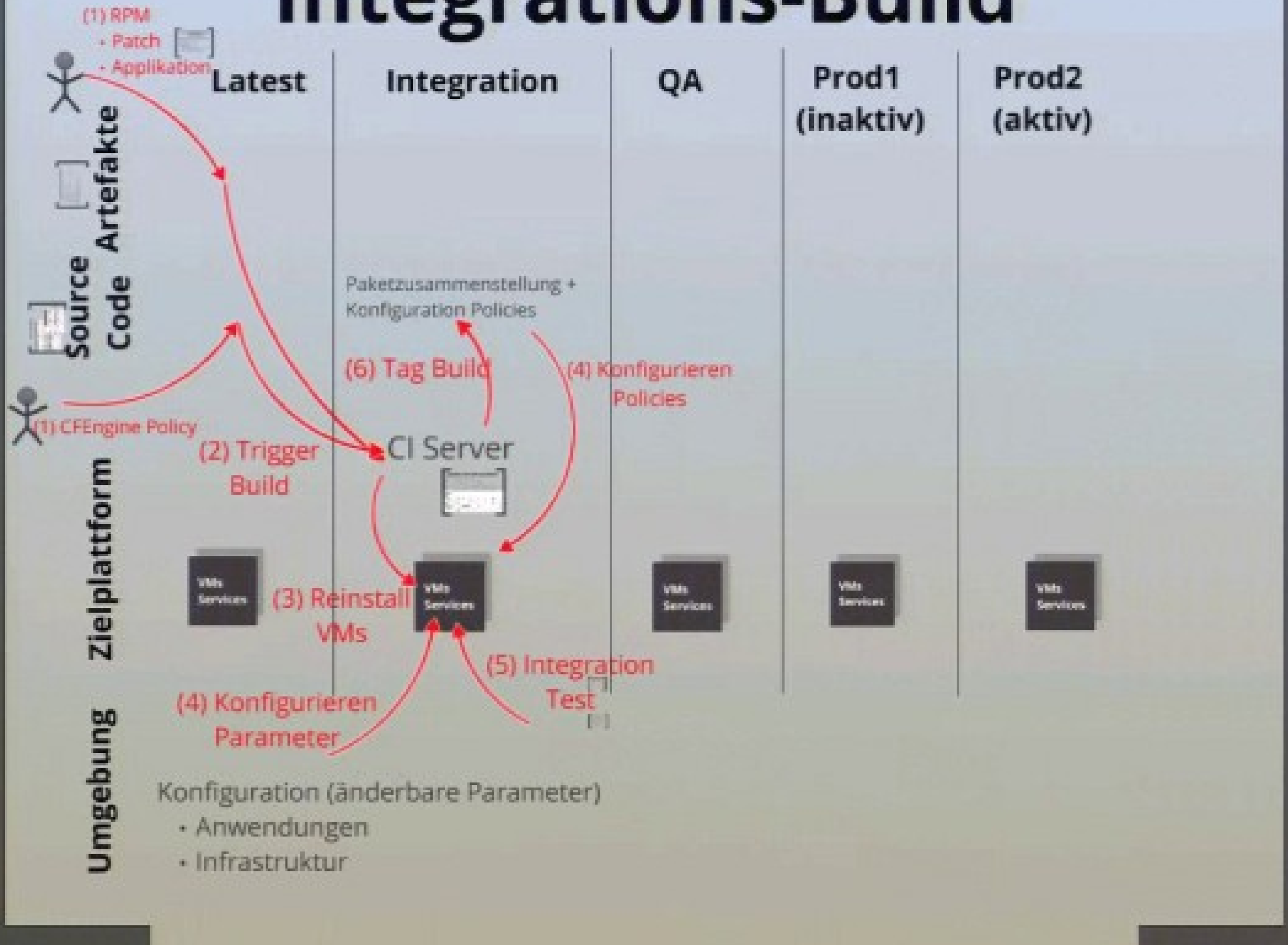
- nehmen die Welt, wie sie ist,
- laufen kontinuierlich,
- finden Störungen, obwohl sich "nichts verändert" hat,
- sind oft Teil einer gemeinsamen Infrastruktur.

Aber:

Beide Verfahren führen Prüfungen/Checks durch und liefern das Ergebnis zurück.

Gemeinsamkeiten nutzen, gemeinsame Testfälle nutzen, z.B. Fitnesse-Tests mit Icinga steuern.

Integrations-Build



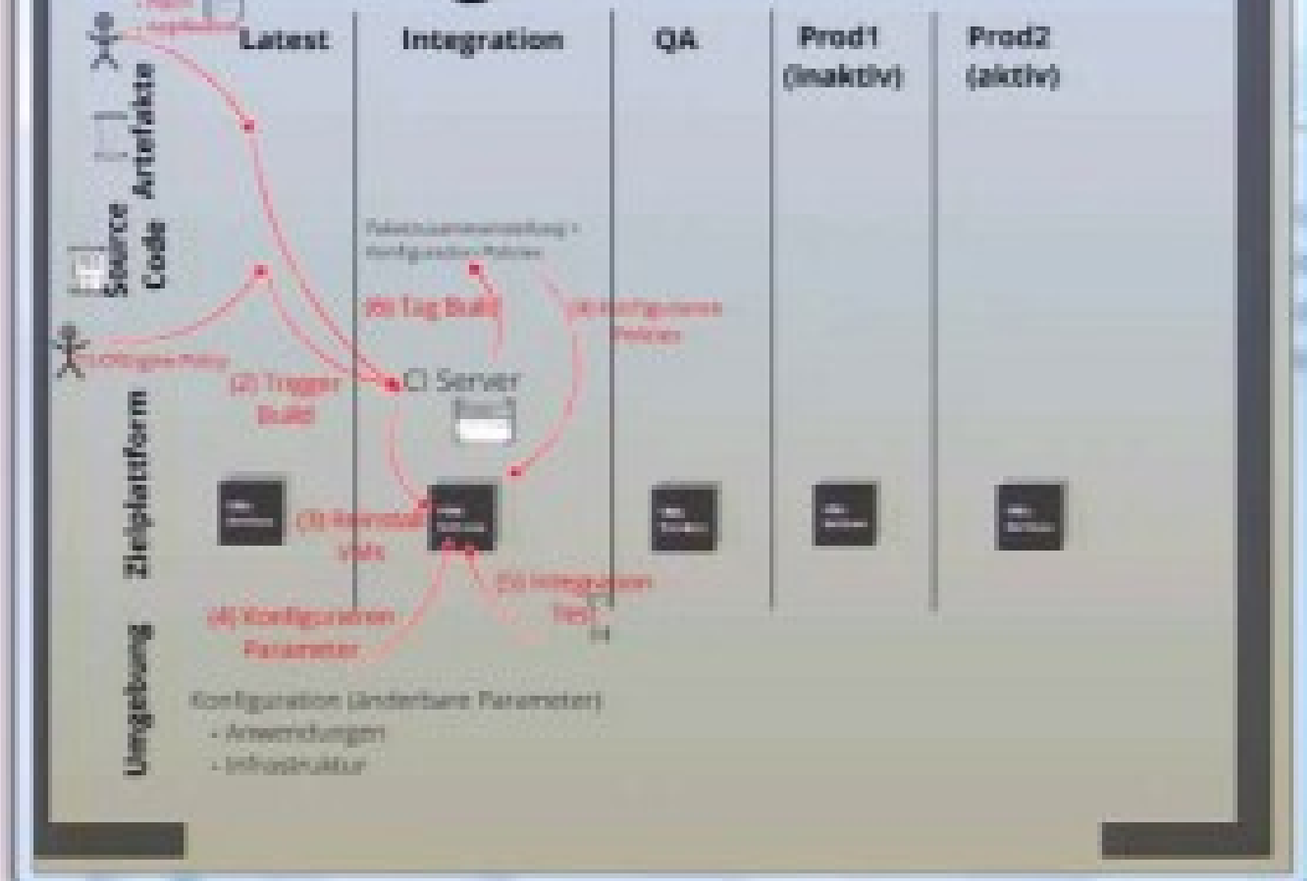
Konfiguration (änderbare Parameter)

- Anwendungen
- Infrastruktur

Adobe Flash Player 10

Datei Ansicht Steuerung Hilfe

Integrations-Build



Suchen

Größe

ordner	
ordner	
endung	4.431 KB
	1 KB

Kontakte
Links
Musik
RZ-Automatisierur
Suchvorgänge
Videos
Öffentlich

prezi
Anwendung
Änderungsdatum: 14.06.2012 20:50
Größe: 4,32 MB
Erstelldatum: 15.06.2012 08:45

RZ-Automatisierung

Adobe Flash Player 10

DE

Big Tech Day ▸ RZ-Automatisierung ▸

Suchen

Organisieren | Ansichten | Öffnen | Freigeben | Brennen

Linkfavoriten
- Dokumente
- Bilder
- Musik
- Weitere »

Ordner
- Big Tech Day
  - Bilder
  - btd5-bigdata
  - Desktop
  - Dokumente
  - Download
  - Favoriten
  - Gespeicherte Spiel
  - Kontakte
  - Links
  - Musik
  - RZ-Automatisierur
  - Suchvorgänge
  - Videos
- Öffentlich

Name	Änderungsdatum	Typ	Größe
data	15.06.2012 08:45	Dateiordner	
prezi-app	15.06.2012 08:45	Dateiordner	
prezi	14.06.2012 20:50	Anwendung	4.431 KB
pwd	14.06.2012 21:35	Datei	1 KB

prezi
Anwendung
Änderungsdatum: 14.06.2012 20:50
Größe: 4,32 MB
Erstelldatum: 15.06.2012 08:45

RZ-Automatisierung

DE

Adobe Flash Player 10

Datei Ansicht Steuerung Hilfe

Prezi

Suchen

Größe

4.431 KB

1 KB

Musik

RZ-Automatisierur

Suchvorgänge

Videos

Öffentlich

prezi
Anwendung

Änderungsdatum: 14.06.2012 20:50
Größe: 4,32 MB
Erstelldatum: 15.06.2012 08:45

RZ-Automatisierung

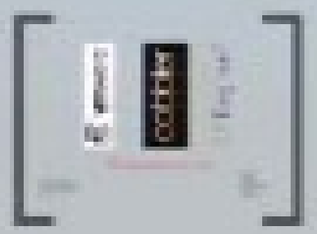
Adobe Flash Player 10

DE

TNG TECHNOLOGY CONSULTING

Virtualisierung und Automatisierung - wie man sich ein Rechenzentrum programmiert

Infrastructure as Code



Fazit

Werkzeuge

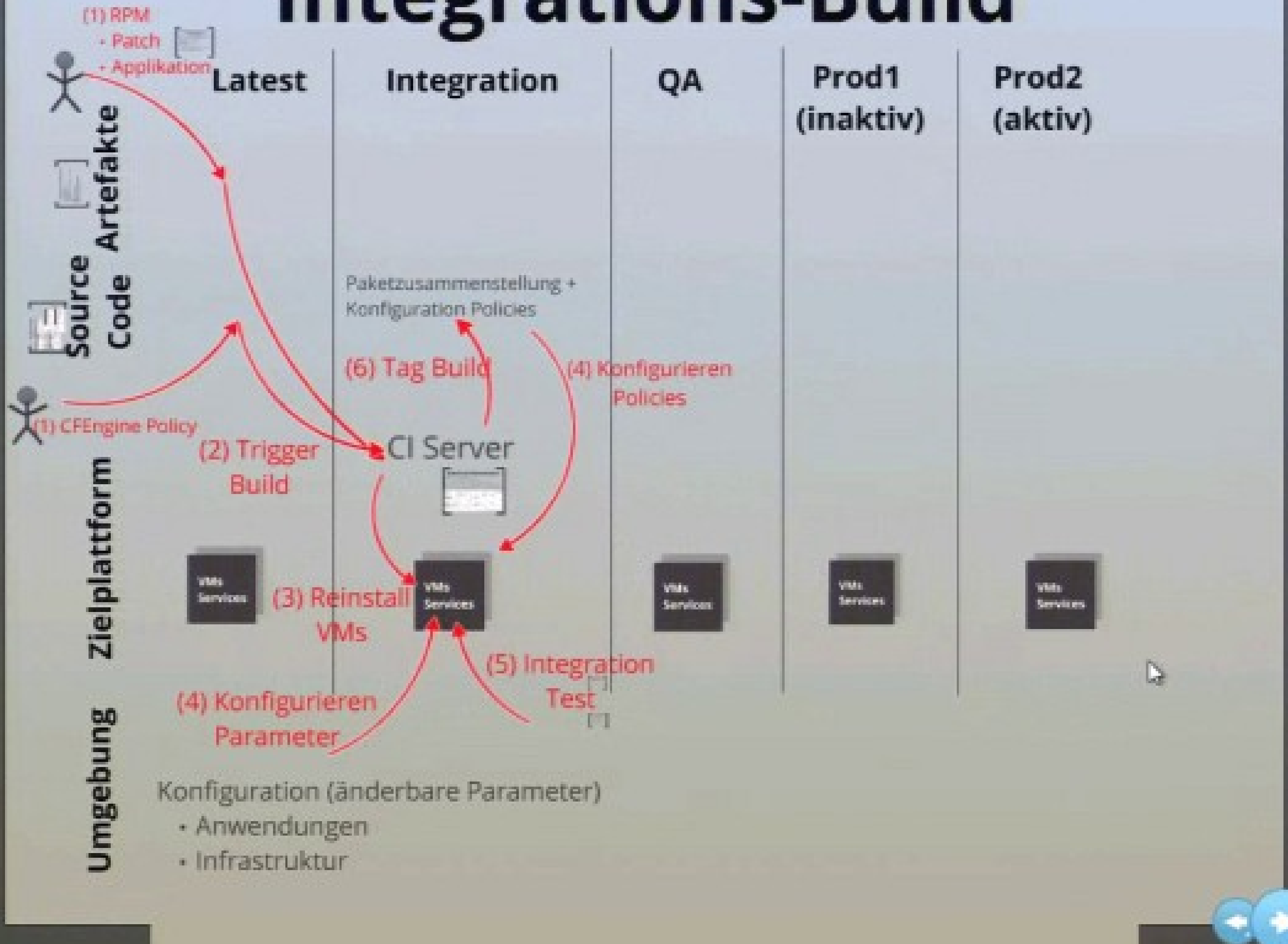
Continuous Delivery

Ressourcen Planung, Virtualisierung

PXE Boot, DHCP Netzwerkinstallation

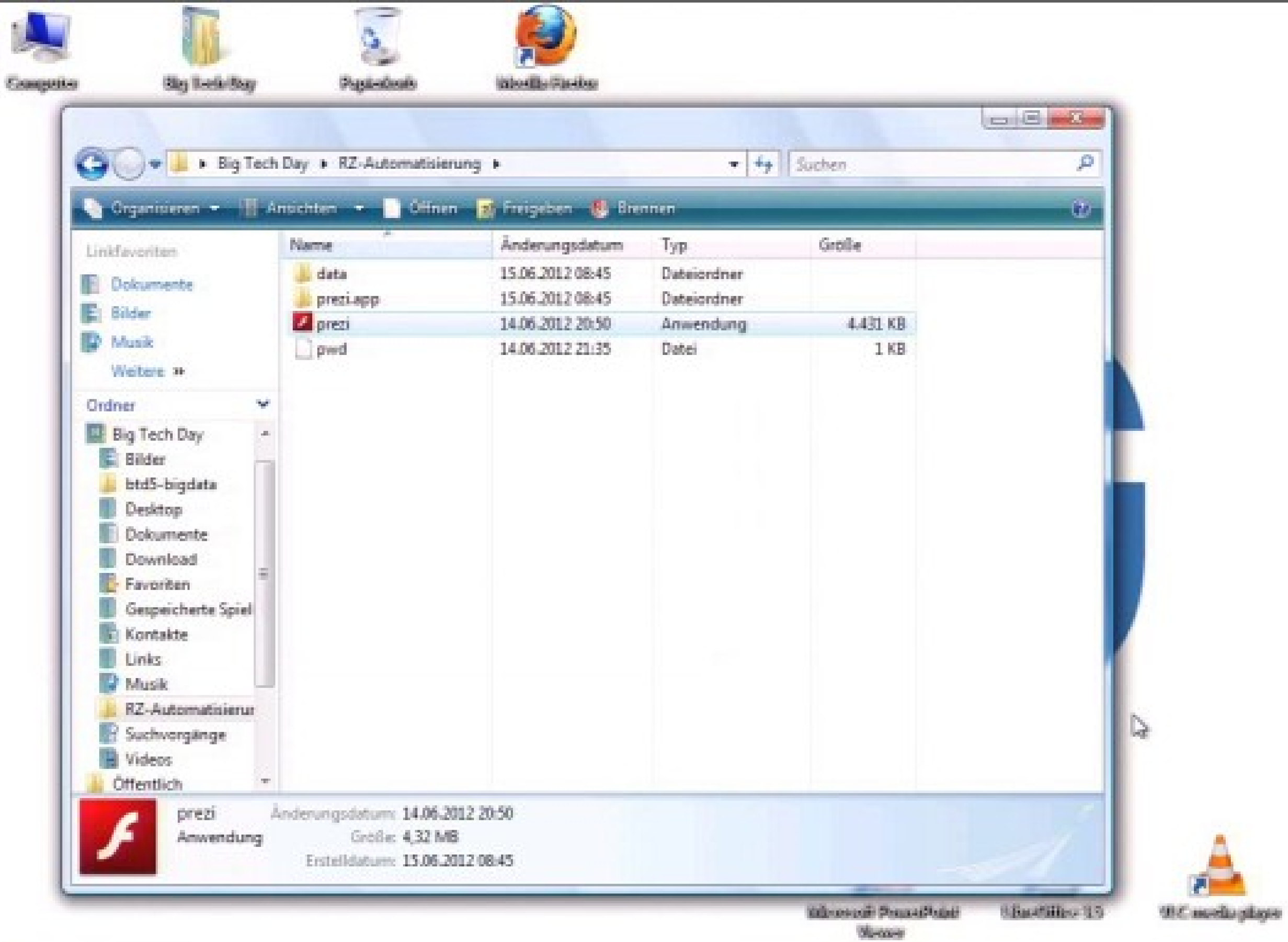
Service + Stage Datei Konfiguration

Integrations-Build



Konfiguration (änderbare Parameter)

- Anwendungen
- Infrastruktur


Big Tech Day ▸ RZ-Automatisierung ▸
Suchen
Organisieren
Ansichten
Öffnen
Freigeben
Brennen
Linkfavoriten
Dokumente
Bilder
Musik
Weitere »
Ordner
Big Tech Day
Bilder
btd5-bigdata
Desktop
Dokumente
Download
Favoriten
Gespeicherte Spiel
Kontakte
Links
Musik
RZ-Automatisierur
Suchvorgänge
Videos
Öffentlich
Name
Änderungsdatum
Typ
Größe
data 15.06.2012 08:45 Dateiordner
prezi-app 15.06.2012 08:45 Dateiordner
prezi 14.06.2012 20:50 Anwendung 4.431 KB
pwd 14.06.2012 21:35 Datei 1 KB
prezi Änderungsdatum: 14.06.2012 20:50
Anwendung Größe: 4,32 MB
Erstelldatum: 15.06.2012 08:45
RZ-Automatisierung
DE
16:17

Big Tech Day ▸ RZ-Automatisierung ▸

Suchen

Organisieren | Ansichten | Öffnen | Freigeben | Brennen

Linkfavoriten
- Dokumente
- Bilder
- Musik
- Weitere »

Ordner
- Big Tech Day
  - Bilder
  - btd5-bigdata
  - Desktop
  - Dokumente
  - Download
  - Favoriten
  - Gespeicherte Spiel
  - Kontakte
  - Links
  - Musik
  - RZ-Automatisierur
  - Suchvorgänge
  - Videos
- Öffentlich

Name	Änderungsdatum	Typ	Größe
data	15.06.2012 08:45	Dateiordner	
prezi-app	15.06.2012 08:45	Dateiordner	
prezi	14.06.2012 20:50	Anwendung	4.431 KB
pwd	14.06.2012 21:35	Datei	1 KB

prezi
Anwendung
Änderungsdatum: 14.06.2012 20:50
Größe: 4,32 MB
Erstelldatum: 15.06.2012 08:45

RZ-Automatisierung

DE

Big Tech Day

Organisieren | Ansichten | Explorer | Freigeben | Brennen

Suchen

Linkfavoriten
- Dokumente
- Bilder
- Musik
- Weitere »

Ordner
- Big Tech Day
  - Bilder
  - btd5-bigdata
  - Desktop
  - Dokumente
  - Download
  - Favoriten
  - Gespeicherte Spi
  - Kontakte
  - Links
  - Musik
  - RZ-Automatisier
  - Suchvorgänge
  - Videos
- Öffentlich

Name	Änderungsdatum	Größe	Ordnerpfad
Bilder	Dateiordner		
btd5-bigdata	Dateiordner		
Desktop	Dateiordner		
Dokumente	Dateiordner		
Download	Dateiordner		
Favoriten	Dateiordner		
Gespeicherte Spiele	Dateiordner		
Kontakte	Dateiordner		
Links	Dateiordner		
Musik	Dateiordner		
RZ-Automatisierung	Dateiordner		
Suchvorgänge	Dateiordner		
Videos	Dateiordner		
BTD5-Scrum-in-Large-Proje...	Adobe Acrobat Document	2,33 MB	
BTD5-Scrum-in-Large-Proje...	Microsoft PowerPoint Prese...		
e-volo Vortrag 2012	OpenDocument Präsentation		

RZ-Automatisierung
Dateiordner
Änderungsdatum: 15.06.2012 08:45

Big Tech Day

DE

Big Tech Day

Organisieren | Ansichten | Explorer | Freigeben | Brennen

Suchen

Linkfavoriten
- Dokumente
- Bilder
- Musik
- Weitere »

Ordner
- Big Tech Day
  - Bilder
  - btd5-bigdata
  - Desktop
  - Dokumente
  - Download
  - Favoriten
  - Gespeicherte Spi
  - Kontakte
  - Links
  - Musik
  - RZ-Automatisier
  - Suchvorgänge
  - Videos
- Öffentlich

Name	Änderungsdatum	Größe	Ordnerpfad

- Desktop — Dateiordner
- Dokumente — Dateiordner
- Download — Dateiordner
- Favoriten — Dateiordner
- Gespeicherte Spiele — Dateiordner
- Kontakte — Dateiordner
- Links — Dateiordner
- Musik — Dateiordner
- RZ-Automatisierung — Dateiordner
- Suchvorgänge — Dateiordner
- Videos — Dateiordner
- BTD5-Scrum-in-Large-Proje... — Microsoft PowerPoint Prese... — 1,03 MB
- BTD5-Scrum-in-Large-Proje... — Adobe Acrobat Document — 2,33 MB
- e-volo Vortrag 2012 — OpenDocument Präsentation — 50,8 MB
- TNG Tech Day_CIOs Corner_Telefonica Germany... — Adobe Acrobat Document

RZ-Automatisierung
Dateiordner
Änderungsdatum: 15.06.2012 08:45

Big Tech Day

BTD5-Scrum-in-Large-Projects.pdf - Adobe Reader

File Edit View Window Help

1 / 40 73,7% Tools Sign Comment

TNG TECHNOLOGY CONSULTING

Scrum in Large Projects

Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger

Big Tech Day BTD5-Scrum-in-Lar... DE 14:48

TNG TECHNOLOGY CONSULTING

Scrum in Large Projects

Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger

TNG TECHNOLOGY CONSULTING

Scrum in Large Projects

Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger

TNG TECHNOLOGY CONSULTING

Scrum in Large Projects

Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger