Continuous Everything

Development, Integration, Deployment, DevOps

Peter Hormanns
cusy GmbH, Berlin
Vortrag | OpenRheinRuhr | 5./6. November 2016
de.slideshare.net/cusyio/continuous-everything

Herzlich Willkommen

Peter Hormanns

- Java Developer
- DevOps
- Free Software Consultant

cusy.io

- SaaS - Software as a Service
- Spezialisierung auf Entwickler-Werkzeuge
- Datenschutz nach deutschem / europäischem Recht

Agenda

- Vorstellung und Agenda
- Die Idee Continuous Delivery und Continuous Deployment
- Werkzeuge für Continuous Delivery und Continuous Deployment
- Die Werkzeuge im Beispielprojekt
- Diskussion

Motivation

Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufriedenzustellen.

Erstes Prinzip des agilen Manifests (2001)

Herausforderungen

- Development ← vs. → Operations
- kontinuierliche Änderung ← vs. → Stabilität
- Termine
  - Iterationen (2 pro Monat)
  - Releases (2 pro Jahr)
- Monitoring
  - der Infrastruktur ✔
  - der Funktionalität ☹

DevOps: Continuous Everything

Agile Entwicklung

DevOps: Continuous Everything

Agile Entwicklung

Continuous Integration

DevOps: Continuous Everything

Agile Entwicklung

Continuous Integration

Continuous Delivery

DevOps: Continuous Everything

Agile Entwicklung

Continuous Integration

Continuous Delivery

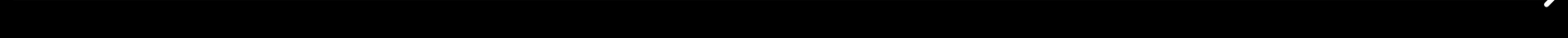
Continuous Deployment

DevOps: Continuous Everything

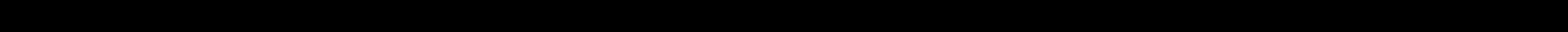
Agile Entwicklung

Continuous Integration

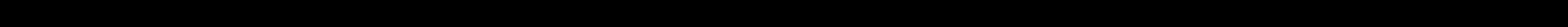
Continuous Delivery



Continuous Deployment



Continuous Configuration Automation



Worum geht es?

In einem Continuous Integration Projekt mergen und commiten alle Entwickler kontinulierlich ihren Arbeitsfortschritt in einen gemeinsamen Branch.

CI-Werkzeuge stellen sicher, dass die Änderungen valide und releasefähig sind.

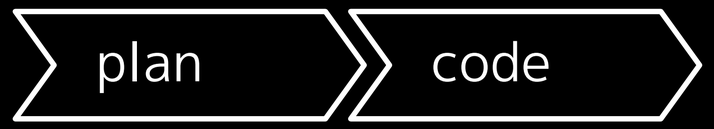
Continuous Integration Regeln

- jeder Entwickler integriert seine Änderungen mindestens täglich
- jeder Push wird durch Build und Tests verifiziert
- unfertiger Code bleibt aussen vor (Developer- oder Feature-Branch)
- fehlerhafte Builds werden sofort repariert oder die Änderung wird verworfen

DevOps Toolchain

- Projektmanagement, Backlog
- Dokumentation, Wiki

DevOps Toolchain

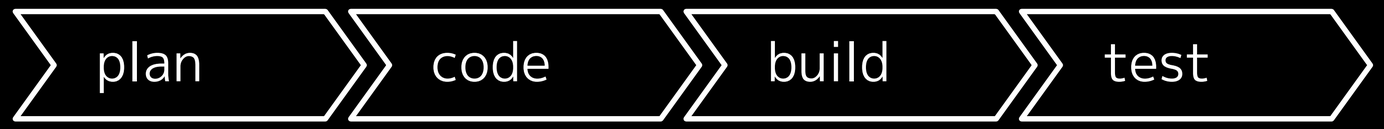


- Entwicklung und Code-Review
- Versionskontrolle

DevOps Toolchain

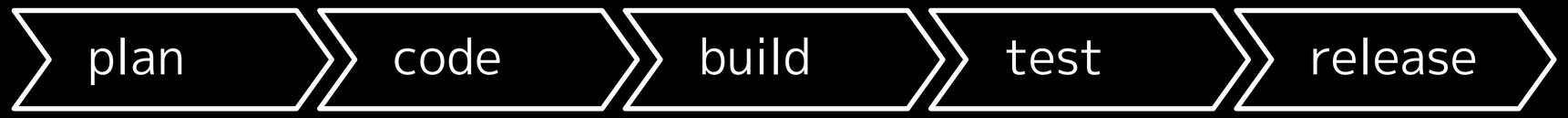
- Continuous Integration, Build-Werkzeuge
- Developer-Test-Automatisierung

DevOps Toolchain



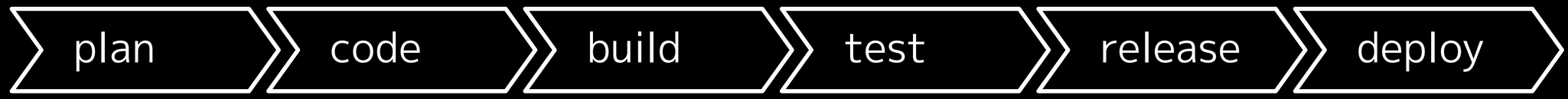
- Continuous-Integration, Testautomatisierung
- Integrationstests
- Akzeptanztests

DevOps Toolchain



- Paketierung
- Pre-Deployment-Staging
- Release-Automatisierung

DevOps Toolchain



- Konfiguration
- Production-Staging
- Infrastruktur als Code

DevOps Toolchain

- Logging
- Exception-Handling
- Performance-Monitoring
- Service-Desk

cusy Lifecycle Werkzeuge

Projekt-management

Jira Software

Versions-verwaltung

Gitblit

Build

Continuous Integration

Release

Deploy

Jenkins

Helpdesk

Jira Service Desk

Dokumenten-management

Confluence

Code Review

Gerrit

Configuration

Ansible

Log-Management & Analyse

Sentry

Webanalyse

Piwik

Unsere Situation

- Am Projekt arbeitet ein kleines verteiltes Team (zwei bis drei Entwickler)
- mehrere kleine Software-Komponenten
- wir bieten eine DevOps Plattform an

  → eat your own dog food

Anforderungen an Continuous Integration

- Alle Änderungen nachvollziehbar
- Einfacher Git-Workflow
- Automatische Tests
  - Entwickler- (=Unit-)Tests
  - Integrations-/Akzeptanztest

Anforderungen an Continuous Delivery

- Release-Tagging im Git
- automatisiertes Packaging
- automatisierte Konfiguration
- automatisiertes Deployment

Lösungen

- Git Self-Hosting Gitblit mit grafischer Oberfläche
- Maven Projekt- und Build-Tool mit Plugins
- Jenkins Contiuous Integration Server
- Ansible als Deployment-Automatisierung für DevOps

Gitblit

Freie Alternative („self-hosted“) zu Github

- 2005 Linux Kernel Entwickler entwickeln Git als Alternative zu BitKeeper
- 2008 Github wird gegründet
- 2009 JGit - Git Implementierung in reinem Java
- seit 2011 kontinulierliche Entwicklung von Gitblit durch James Moger
- Alternativen Gitolite, Gitlab, Gogs

Maven

Java Build Automation

- 2000 Apache Ant, eine Art „make für Java-Projekte“
- 2004 erstes Maven Release
  - vorgegebene Projektstruktur
  - Verwaltung von Abhängigkeiten
  - Repositories für Build-Artefakte
  - Plugin-Architektur

Jenkins CI

Continuous Integration

- 2001 Agiles Manifest: Continuous Integration
- 2005 Hudson 1.0
- 2010 im Januar übernimmt Oracle Sun Microsystems
- 2010/2011 Jenkins forks
- 2016 Jenkins 2.0
- Alternativen: Gitlab CI, Travis CI (Webservice), Bamboo (proprietär)

Ansible

Configurationmanagement, Infrastructure as Code

- 1993 CFEngine
- 2005 Puppet
- 2012 Ansible: „DevOps“ ohne root-Rechte

Continuous Configuration Automation

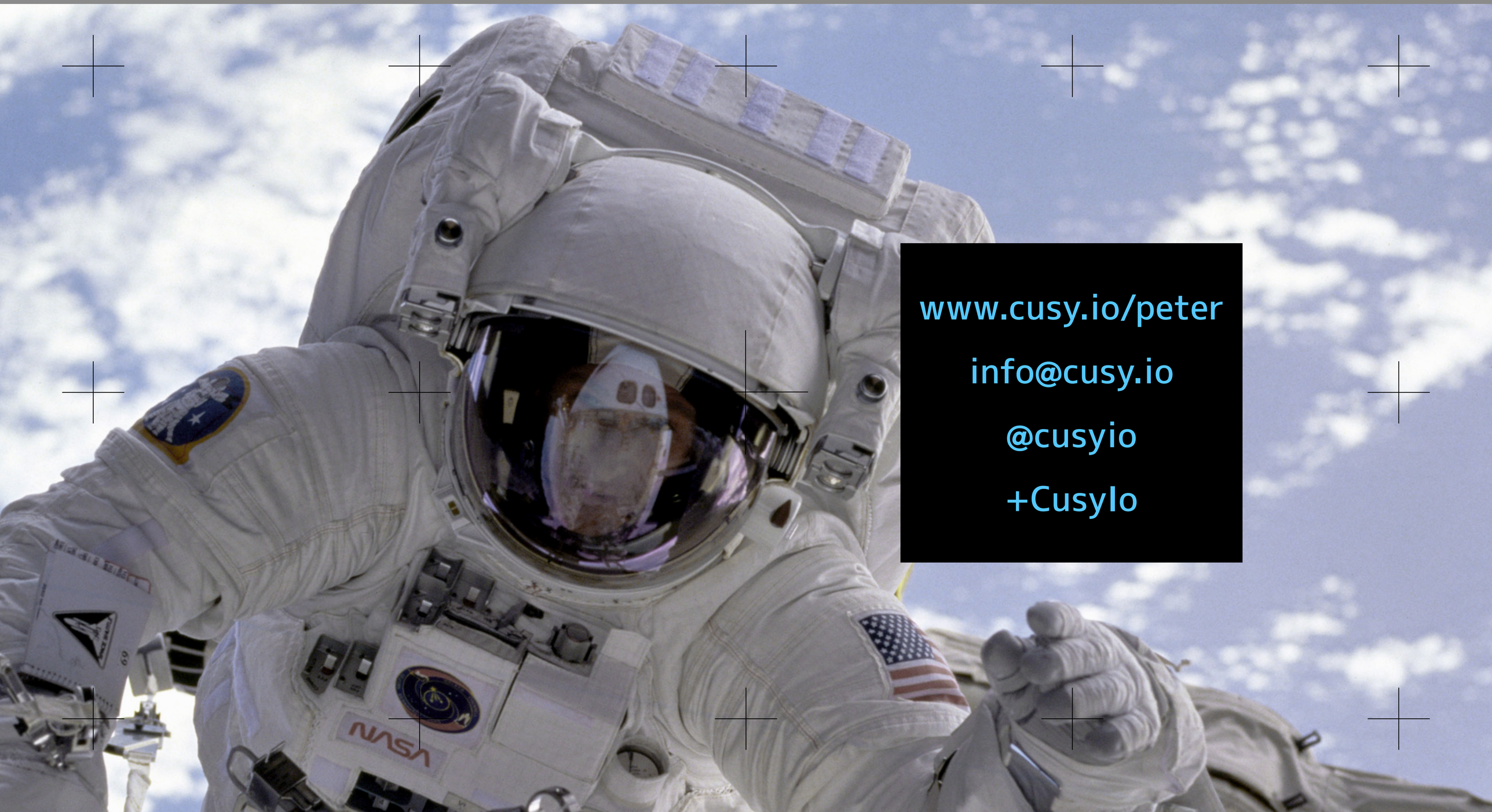
Continuous Configuration Automation

Die Werkzeuge in der Praxis (Beispielprojekt)

Fragen und Diskussion

Ich freue mich auf Eure Fragen, Ideen und Diskussionsbeiträge.

Kontakt



www.cusy.io/peter

info@cusy.io

@cusyio

+CusyIo

Bildnachweise

- droidcon Berlin 2015 – Hackathon; cc BY-SA 2.0: droidcon Global
- Peter Hormanns
- Cusy-Messestand; Veit Schiele
- Michael Gernhardt in space during STS-69 in 1995; PUBLIC DOMAIN: NASA