

Seminararbeit im Studiengang WInf (BSc)

# Increasing developer productivity through dev ops continuous integration process illustrated using jenkins

Version 0.1.0 vom 15. November 2016  
(Vor Abgabe entfernen)

**Anzumana Taal**

283400

anzumana.taal@me.com

**Betreuer:**

Prof. Dr. Klaus Schmid, SSE

## Eigenständigkeitserklärung

### **Erklärung über das selbstständige Verfassen von "Increasing developer productivity through dev ops continuous integration process illustrated using jenkins"**

Ich versichere hiermit, dass ich die vorstehende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der obigen Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, habe ich in jedem Fall durch die Angabe der Quelle bzw. der Herkunft, auch der benutzten Sekundärliteratur, als Entlehnung kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet und anderen elektronischen Text- und Datensammlungen und dergleichen. Die eingereichte Arbeit ist nicht anderweitig als Prüfungsleistung verwendet worden oder in deutscher oder einer anderen Sprache als Veröffentlichung erschienen. Mir ist bewusst, dass wahrheitswidrige Angaben als Täuschung behandelt werden.

Hildesheim, den 15. November 2016

---

Anzumana Taal

---

## **Kurzfassung**

Eine kurze Zusammenfassung der Arbeit, die Interesse beim Leser wecken soll.

## **Abstract**

Gerne zusätzlich oder alternativ in Englisch.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>ii</b>
<b>Tabellenverzeichnis</b>	<b>ii</b>
<b>Liste der Listings</b>	<b>iii</b>
<b>Abkürzungsverzeichnis</b>	<b>iv</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Ziel der Arbeit . . . . .	1
1.3. Aufbau der Arbeit . . . . .	1
<b>2. Ebenen</b>	<b>2</b>
2.1. Abschnitt . . . . .	2
2.1.1. Unterabschnitt . . . . .	2
2.1.1.1. Unter-Unterabschnitt . . . . .	2
<b>3. testname</b>	<b>4</b>
3.1. Beispiele . . . . .	4
<b>4. What is Jenkins?</b>	<b>5</b>
4.1. Jenkins . . . . .	5
4.2. Plugins . . . . .	5
4.2.1. Plugin:Blue Ocean beta . . . . .	5
4.2.2. Caveat . . . . .	5
4.2.3. Limitations . . . . .	5
4.3. Pipelines . . . . .	6
<b>5. Schlussbemerkungen</b>	<b>7</b>
5.1. Fazit . . . . .	7
5.2. Ausblick . . . . .	7
<b>A. Anhang</b>	<b>8</b>
A.1. Beispiele . . . . .	8
<b>Glossary</b>	<b>10</b>
<b>Literaturverzeichnis</b>	<b>11</b>

## Abbildungsverzeichnis

A.1. Das Logo der SUH . . . . .	8
---------------------------------	---

## Tabellenverzeichnis

4.1. Source Code Languages . . . . .	5
A.1. Eine Tabelle . . . . .	8

## Liste der Listings

4.1. sample Pipeline . . . . .	6
A.1. HelloWorld . . . . .	9
A.2. Beispiel eines Log-Eintrags . . . . .	9

## Abkürzungsverzeichnis

API	Application Programming Interface
XML	Extensible Markup Language

# **1 Einleitung**

FließtextFließtextFließtextFließtext

## **1.1 Motivation**

FließtextFließtextFließtextFließtext

## **1.2 Ziel der Arbeit**

FließtextFließtextFließtextFließtext

## **1.3 Aufbau der Arbeit**

Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext  
Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext  
Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext Fließtext  
Fließtext Fließtext Fließtext Fließtext Fließtext



## 2 Ebenen

In diesem Kapitel wird kurz aufgezeigt, welche Gliederungsebenen es gibt. Hier befinden wir uns auf der Ebene eines **Kapitels**.

### 2.1 Abschnitt

#### 2.1.1 Unterabschnitt

##### 2.1.1.1 Unter-Unterabschnitt

Soll noch tiefer gegliedert werden ist eine Änderung der Formatierungen und/oder das Verwenden eines zusätzlichen Paketes notwendig.

Increasing developer productivity through dev ops continuous integration process illustrated using jenkins

## **3    testname**

### **3.1   Beispiele**

Zitat ohne Seitenangabe: [5]

bookel [2]

## 4 What is Jenkins?

The following chapter will provide a quick introduction to Jenkins and what it can do for any continuous integration efforts.

### 4.1 Jenkins

There exists a multitude of software solutions that aid in the process of continuous integration. One of those tools is Jenkins.

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project. [5]

We will be using Jenkins to better illustrate how the different areas of the continuous integration process can be modelled and made available in Jenkins. Being an open source project provides us with the option to highlight part of the source code, where we need to. The source itself is written mostly in Java as can be seen in Table 4.1. We assume the reader has a basic understanding of programming in Java. And should be familiar with Javascript, HTML, CSS. The Jenkins software is being released under the MIT License [4], this ensures that our findings will be verifiable by anyone if they so desire.

Tabelle 4.1.: Source Code Languages

Language	Percentage
Java	81.9
HTML	9.8
Javascript	3.4
CSS	2.9
Groovy	1.5
Ruby	0.2
Other	0.3

### 4.2 Plugins

To extend the Jenkins software, the architecture enables developers to create Plugins. By default Jenkins offers an infrastructure to create and publish those plugins.

#### 4.2.1 Plugin:Blue Ocean beta

The user interface got a complete redesign as part of the Blue Ocean Project.[3] Most illustrations will be using this plugin when showing the Jenkins interface.

#### 4.2.2 Caveat

Some plugins will have a negative effect on the performance of Jenkins please keep this in mind when trying to replicate our findings.

#### 4.2.3 Limitations

The plugin architecture is not well suited to performance benchmark our continuous integration process.[6]

## 4.3 Pipelines

Pipelines are Jenkins approach to enable Continuous Delivery. Referring back to our chapter on Continuous Delivery, Pipelines enable Jenkins to support the entire application lifecycle. By creating plain text files in a domain specific language whose syntax is roughly based on Groovy. See 4.1

```
1 parallel 'integration-tests':{  
2     node('mvn-3.3'){}  
3 }, 'functional-tests':{  
4     node('selenium'){}  
5 }
```

Listing 4.1: sample Pipeline

To install Jenkins run

```
1 brew cask install jenkins
```

## **5 Schlussbemerkungen**

FließtextFließtextFließtextFließtext

### **5.1 Fazit**

FließtextFließtextFließtextFließtext

### **5.2 Ausblick**

FließtextFließtextFließtextFließtext

## A Anhang

### A.1 Beispiele

Zitat ohne Seitenangabe: [1]

Zitat mit Seitenangabe: [1, S.1]

Referenz eines Glossareintrags: Computer

Eine normale Liste:

- Ein Punkt
- Ein anderer Punkt

Eine nummerierte Liste:

1. Erstens
2. Zweitens
3. ...

Eine Tabelle:

Tabelle A.1.: Eine Tabelle

Spalte 1	Spalte 2	Spalte 3
1	2	3

**Fetter Text**

*Kursiver Text*

Eine Referenz: Siehe Tabelle A.1: Eine Tabelle auf Seite 8.

Ein Bild:



Abbildung A.1.: Das Logo der SUH

Eine abgesetzte Formel:

$$\sum_{n=0}^3 n = 6 \tag{A.1}$$

Eine Formel im Fließtext:  $2^2 = 4$

Ein Listing aus einer Datei:

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello World!");  
4     }  
5 }
```

Listing A.1: HelloWorld

Ein 'on-the-fly' erstelltes Listing:

```
1 (Sun Sep 13 23:02:20 2009): ODBC Driver <system32>\wbemdr32.dll not present  
2 (Sun Sep 13 23:02:20 2009): Successfully verified WBEM ODBC adapter (incompatible version  
   removed if it was detected).  
3 (Sun Sep 13 23:02:20 2009): Wbemupgd.dll Registration completed.
```

Listing A.2: Beispiel eines Log-Eintrags



## Glossar

**Computer** ist ein elektronisches Gerät, das Daten verarbeitet.

## Literaturverzeichnis

- [1] Günther Böckle, Peter Knauber, Klaus Pohl, and Klaus Schmid. *Software-Produktlinien*. Dpunkt-Verlag, 2004.
- [2] Günther Böckle, Peter Knauber, Klaus Pohl, and Klaus Schmid. *Software-Produktlinien*. Dpunkt-Verlag, 20098.
- [3] Jenkins. <https://jenkins.io/projects/blueocean/>, 2008.
- [4] Jenkins. Mit license, 2008.
- [5] jenkins incorpora. my title for the docs, 2008.
- [6] Jan Waller, Nils C. Ehmke, and Wilhelm Hasselbring. Including performance benchmarks into continuous integration to enable DevOps. *ACM SIGSOFT Software Engineering Notes*, 40(2):1–4, apr 2015.