

# **Exposé zur Bachelor-Thesis**

**Gestaltung eines Softwareentwicklungsprozess und  
exemplarische Durchführung zur  
Geschäftsprozessautomatisierung in jungen Unternehmen  
am Beispiel der Crowdcode GmbH & Co. KG**

Verfasser: Nooshin Naghavi  
Martikelnnummer: 2401101

1. Betreuer: Dr. Günter Kniesel
2. Betreuer: Ingo Düppe

23. Januar 2016

## 1 Motivation

Unternehmen setzen eine Vielfalt von Werkzeugen und Notationen zur Modellierung und Optimierung von Geschäftsprozessen ein. Eine Standardsprache zur Prozessmodellierung ist BPMN (Business Process Model and Notation) [1], welche zwischenzeitlich von OMG (Object Management Group) übernommen, standardisiert und weiter entwickelt wurde. [2] Die Symbole in BPMN sind einfach gestaltet und erleichtern die Kommunikation zwischen Kunden und IT-Fachleuten. Dadurch können die Anforderungen schneller modelliert werden.

Die von BPMN gestalteten Prozessen können mittels BPMN-Engines ausgeführt werden und Camunda BPM [3] ist eine davon. Im Gegensatz zu klassischen BPM-Suiten ist Camunda ein offenes Framework, das in der vorhandenen technischen Umgebung nahtlos eingebettet werden kann. [4]

## 2 Zielsetzung und Grenzen

### 2.1 Ausgangssituation

Crowdcode GmbH & Co. KG wurde im September 2014 von Ingo Düppe und Marcus Nörder-Tuitje gegründet. Crowdcode ist in dem Bereich der Gestaltung und Steuerung von Softwareentwicklungsprozessen sowie der Entwicklung von Unternehmensanwendungen tätig. Spezialisiert hat sich Crowdcode auf Java EE-Technologien. Zurzeit beschäftigt Crowdcode 9 IT-Berater und Softwareentwickler für die Durchführung von Softwareentwicklungsprojekten.

Crowdcode Geschäftsbereiche sind Beratung und Unterstützung bei der Softwareentwicklung von Unternehmensanwendungen, Schulungen und Coaching, Entwicklung von SaaS-Produkten und Open Source Entwicklung.

Für das weitere Wachstum wurde die strategische Entscheidung getroffen, dass die existierenden und auch zukünftigen Geschäftsprozesse weitestgehend automatisiert werden sollen.

### 2.2 Geschäftsprozesse

Mögliche Geschäftsprozesse kommen aus den Bereichen:

- Fakturierung
- Buchhaltung
- Reisekostenabrechnung
- Monatliche Umsatzsteuermeldung
- Belegmanagement
- Projektreporting
- Belegmanagement

Ein wesentlicher Geschäftsprozess von Crowdcode ist die monatliche Abrechnung gegenüber Kunden und Subunternehmern. In den Projekten werden Zeitdaten von Mitarbeitern und Subunternehmern erfasst, welche die Grundlage der monatlichen Rechnungen für die Kunden darstellen. Neben der Generierung von Projektreports über die geleisteten Arbeiten und deren Aufwand sind die aufgezeichneten Zeitdaten zu aggregieren, manuell zu prüfen und den Kunden als Projektreport automatisch zur Verfügung zu stellen. Gleichfalls sind Zeitdaten mit den Rechnungen der jeweiligen Subunternehmern zu prüfen. Rechnungen der Subunternehmer sollen automatisch per E-Mail oder über einen Microservice eingereicht werden. Rechnungen der Kreditoren sind entsprechend freizugeben und automatisch zum Ende des Zahlungsziel zu bezahlen.

Zugleich müssen Reisekosten erfasst, geprüft und freigegeben bzw. gegebenenfalls in Rechnung gestellt werden. Sämtliche Zahlungsbelege sollen digitalisiert, archiviert sowie einem Steuerberater zur Verfügung gestellt werden.

### 2.3 Zielsetzung

Ziel der Arbeit ist die prototypische Automatisierung von Geschäftsprozessen mittels einer BPMN-Engine und der Implementierung verschiedener Microservices. Dabei wird besonderes auf Analyse und Modellierung der Geschäftsprozesse eingegangen und der Softwarearchitektur konzeptioniert. Zudem soll eine Continuous Delivery Pipeline entwickelt werden, mittels derer eine stetige evolutionäre Entwicklung der Geschäftsprozesse und der notwendigen Microservices ermöglicht wird.

### 2.4 Softwarearchitektur

Für die Umsetzung der Geschäftsprozesse soll eine Microservice-Architektur entwickelt werden.

Mögliche Technologien sind:

- Java Enterprise Edition [5]
- Spring Framework [5]
- Spring Boot [6]
- Spring Cloud [7]
- Camunda BPM Engine [8]
- Google Web Toolkit [9]
- Docker [10]

### 2.5 Gestaltung des Entwicklungsprozess

Der Entwicklungsprozess soll eine Continuous Delivery Pipeline [11] sein.

Werkzeugkette: - IntelliJ JetBrains - Atlassian BitBucket (Git-Server) - Atlassian Jira (Projektmanagemnet) - Atlassian Confluence (Dokumentaion) - Apache Jenkins (Continuous Integration Server) - SonarQube

Prozess: Agiler Prozess (Scrum / Kanaban)

### 2.6 Zielgruppe

Hauptnutzer der Anwendung werden folgendes sein:

- Geschäftsführung
- Mitarbeiter
- Subunternehmer / Dienstleister
- Kunden
- Steuerberater

## 2.7 Abgrenzungen

- In der Arbeit soll zunächst eine grobe Analyse der Kerngeschäftsprozesse, der involvierten Informationssystemen und der organisationsinternen und -externen Nutzergruppen erfolgen.
- Aufbauend auf dieser Analyse werden einzelne Geschäftsprozesse exemplarisch detaillierter analysiert, modelliert und in Teilen implementiert.
- Eine vollständige Analyse, Modellierung und Umsetzung der Geschäftsprozesse ist im Rahmen dieser Arbeit nicht vorgesehen.
- Im Sinne von Continuous Delivery und einer agilen Entwicklung soll frühzeitig ein minimaler Entwicklungsprozess erstellt werden.

### 3 Gliederung

Die Gliederung kann im Laufe der Arbeit geändert werden.

- Eidesstattliche Erklärung
- Inhaltsverzeichnis
- Abstract
- Einleitung
  - Motivation
  - Ausgangssituation
  - Zielsetzung
  - Aufbau der Arbeit
- Analyse und Design der Geschäftsprozesse und Informationssysteme
  - Beschreibung der Geschäftsprozesse mit BPMN
  - Beschreibung der Informationssysteme
  - Ableitung von Anforderungen für eine Automatisierung
  - Zusammenfassung
- Konzeption der Softwarearchitektur
  - Beschreibung der Softwarearchitektur
    - \* Grundlagen von Service- bzw. resourceorientierten Architekturen
    - \* Process Engine, Configuration Server, Monitoring etc.
  - Zusammenfassung
- Konzeption des Softwareentwicklungsprozess
  - Beschreibung des verwendeten Technologieportfolios
    - \* Git, Jira, Jenkins, Confluence, BPMN Model, MagicDraw, IntelliJ etc.
  - Beschreibung der Continuous Delivery Pipeline
  - Beschreibung der einzelnen Stages (Test, Integration, Akzeptanz, Produktion)
    - \* Automatisierte Tests (Test Driven Development)
    - \* Unit Tests
    - \* Integrationstests
    - \* Automatische Akzeptanztests (wenn noch Zeit ist)
- Entwicklung ausgewählter Geschäftsprozesse
  - Geschäftsprozessmodelle (Detail)
  - Ableitung von Epics, User Stories.
  - Komponentenmodell
  - Beschreibung der einzelnen Microservices (Schnittstellen, Masken, etc.)
  - Zusammenfassung
- Diskussion
  - Auswertung der Ergebnisse
    - Ergebnisse der einzelnen Iterationen
    - Impediments der Iterationen
  - Fazit
- Literaturverzeichnis

## 4 Zeitplanung

Die Arbeit wird in Anlehnung an einem agilen Entwicklungsprozess in 10 Iterationen durchgeführt. Jede Iteration ist zwei Wochen lang und beginnt mit einer Planung des Iterationsziel und der einzelnen Aufgaben, welche hinsichtlich ihrer Priorität sortiert werden. Die Iteration endet mit dem Review der Arbeitsergebnisse und einer Retrospektive. Zur Definition of Done gehört hierbei insbesondere neben der Auslieferung eines funktionsfähigen Prototyps vor allem die zugehörige Ausarbeitung innerhalb der Bachelorarbeit.

Zeitpunkt	Aufgaben und Meilenstein
29.01.16	Abgabe des Exposé
	Beginn der Einarbeitung in relevanten Technologien
15.03.16	Anmeldung zur Bachelorarbeit
	8 Iterationen (zweiwöchig)
15.07.16	Beginn der Pufferzone
	2 Iterationen (zweiwöchig)
15.08.16	Abgabe der Bachelorarbeit

Summe der Bearbeitungszeit: ca. 20 Wochen

Iteration	Ziel
1	Analyse und Dokumentation der Geschäftsprozesse und der verwendeten Informationssysteme. Modellierung der relevanten Geschäftsprozesse, Erfassung von Anforderungen und ableiten von Epics und User Stories.
2	Konzeption und Dokumentation der Softwarearchitektur auf Basis von Spring Boot Microservices und GWT.
3	Konzeption und Dokumentation des Softwareentwicklungsprozess. Konfiguration der Continuous Delivery Pipeline und automatische Unit, Integrationstests. Automatische Deployments in die Integrationsumgebung.
4	Detailanalyse ausgewählter Geschäftsprozesse. Modellierung notwendiger Microservices. Mockups für benötigte Webanwendungen.
5	Analyse, Design, Implementierung und Dokumentation
6	Analyse, Design, Implementierung und Dokumentation
7	Analyse, Design, Implementierung und Dokumentation
8	Analyse, Design, Implementierung und Dokumentation
9	Evaluation / Diskussion der Iterationen und Arbeit
10	Puffer für die Überarbeitung der Arbeit

## 5 Literatur

### Literatur

- [1] OMG (2011): Business Process Model and Notation (BPMN). URL:<http://www.omg.org/spec/BPMN/2.0/>. Stand: 22.01.2016.
- [2] Allweyer, Thomas (2015): BPMN 2.0 Einführung in den Standard für die Geschäftsprozessmodellierung, 3. Auflage. Norderstedt, S. 11.
- [3] Camunda, <https://camunda.org/features/engine/>. Stand: 22.01.2016.
- [4] Camunda the business process company (2015): Camunda BPM im Vergleich mit alternativen. URL:<https://network.camunda.org/whitepaper/7>. Stand: 22.1.2016.
- [5] Oracle: Java Platform, Enterprise Edition (Java EE) 7. URL:<http://docs.oracle.com/javaee/7/index.html>. Stand: 19.01.2016.
- [6] Johnson, Hoeller, Donald, Sampaleanu, Harrop, Risberg, Arendsen, Davison, Kopylenko, Pollack, Templier, Vervaet, Tung, Hale, Colyer, Lewis, Leau, Fisher, Brannen, Laddad, Poutsma, Beams, Abedrabbo, Clement, Syer, Gierke, Stoyanchev, Webb, Winch, Clozel, Nicoll, Deleuze (2004-2015): Spring Framework Reference Documentation. URL:<http://docs.spring.io/spring/docs/4.3.0.BUILD-SNAPSHOT/spring-framework-reference/htmlsingle/>. Stand: 19.01.2016.
- [7] Webb, Syer, Long, Nicoll, Winch, Wilkinson, Overdijk, Dupuis, Deleuze (2013-2016): Spring Boot Reference Guide. URL:<http://docs.spring.io/spring-boot/docs/1.3.3.BUILD-SNAPSHOT/reference/htmlsingle/>. Stand: 19.01.2016.
- [8] Spring Cloud Config. URL:<http://cloud.spring.io/spring-cloud-config/spring-cloud-config.html>. Stand: 19.01.2016.
- [9] Google Web Toolkit (GWT): Developer's Guide URL: <http://www.gwtproject.org/doc/latest/DevGuide.html>. Stand: 19.01.2016.
- [10] Docker: URL: <https://docs.docker.com/>. Stand: 19.01.2016.
- [11] Humble, J., Farley, D. (2011): Continuous Delivery. Reliable Software Releases Through Build, Test, and Deployment Automation, Boston.