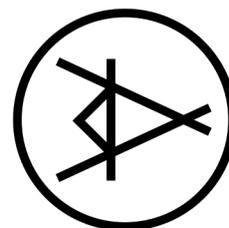


Software-Entwicklung
Micha Barthel
Mühlenstrasse 136 B
28779 Bremen

E-Mail: jujumicha@web.de



Betreff: Referenzen

„Micha Barthel's bemerkenswerte Auffassungsgabe hat QUBUS media bei einem komplizierten Update-Projekt sehr weitergeholfen. Er hat sich fix in unsere über Jahre gewachsene Software zur Verwaltung von Daten und Medien reingearbeitet und die unübersichtliche System-Architektur in dem knappen Zeitfenster aktualisiert und optimiert. Komplexe Aufgaben schrecken ihn nicht ab. Er spricht nicht nur Probleme direkt an - besonders hervorzuheben sind die konstruktiven Lösungsvorschläge und pragmatischen Ideen, die er in die Zusammenarbeit und Kommunikation mit einbringt. Gerne wieder.“

Jessica Weigel | qubus.media

Druckerei und Medienhaus in Hannover.

:// WEBMEN

„Herr Barthel zeichnet sich durch sein ungewöhnlich schnelles Eindringen in komplexe Strukturen und Prozesse aus. Er unterstützt uns immer wieder in Entwicklung, Refactoring und Bug Fixing unserer Java- und Laravel-Projekte und ist für uns ein unverzichtbarer Troubleshooter.“

Christiane Niebuhr-Redder | webmen.de

Webdesign-Agentur aus Bremen für anspruchsvolle Websites & Online-Marketing.

wilde van rhee

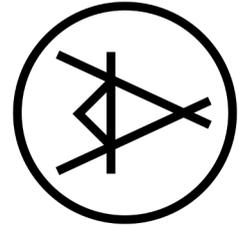
„Micha Barthel hat ohne Vorwissen ein Kundenprojekt auf Magento-Basis mit über eine Millionen Kunden optimiert und als Kernel-Entwickler anspruchsvolle Systemfehler entfernt. Die Planung und Konzeption von komplexer Softwarearchitektur ist seine Leidenschaft.“

Alexander Boroš | wildevanrhee.de

E-Commerce Agentur aus Köln.

Software-Entwicklung
Micha Barthel
Mühlenstrasse 136 B
28779 Bremen

E-Mail: jujumicha@web.de



Betreff: Skills

Personendaten

Name: Micha Barthel
Staatsangehörigkeit: deutsch
EDV-Erfahrung seit: 2003
Einsatzort: Deutschland

Qualifikation

Schwerpunkte: Software-Architektur, Software-Entwicklung
Sprachen: Deutsch / Englisch

Programmierung

Java (Spring): Microservices, Reactive, Cloud, Event-driven, Batch, MVC
PHP: Laravel, Symfony, Magento-Core, Typo3-Core
JavaScript: Vue.js, React, Angular, jQuery
Datenbanken: Oracle (Pl/SQL), MySql, PostgreSql, Hibernate
Web: HTML(5), CSS(3), Bootstrap
Sprachen: Java, PHP, C, C++, C Sharp, Visual Basic, Python, JavaSkript, Kotlin, Shellskripts...
IoT: Raspberry Pi, Arduino, ESP32, Linux-Programmierung

Tools

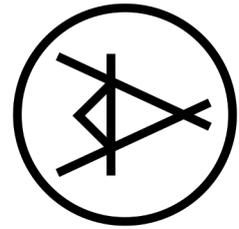
IDE: IntelliJ, Eclipse
Build-System: Maven, Gradle
Deployment / Delivery: Kubernetes, Docker
Application-Server: GlassFish, JBoss, Tomcat
Versionsverwaltung: Git, SVN
Dokumentation: Markdown, Confluence, Wikis, UML
Projektverwaltung: Jira, Trello

Hobbies

Musik: www.miresq.de
Sport: Rad (100-250 Km pro Woche)

Software-Entwicklung
Micha Barthel
Mühlenstrasse 136 B
28779 Bremen

E-Mail: jujumicha@web.de

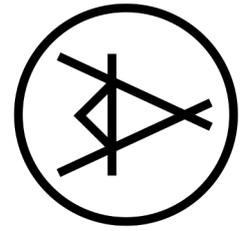


Betreff: Lebenslauf

- Lehrerkind (Grundschullehrerin/Oberstudienrat)
- Realschule
- Selbststudium Informatik in 3 Monaten (siehe Webshop).
- Halbes Jahr Fachinformatiker-Ausbildung bei PantaRhei-Systems (aufgekauft von INIT). Nachdem sich herauskristallisiert hatte, dass ich bereits wichtige Entwickler-Tätigkeiten ausführte, hatte ich mich fest anstellen lassen und die Ausbildung an der Berufsschule abgebrochen.
- Nach einem Jahr gekündigt und in die Selbstständigkeit gegangen für 1-2 Jahre.
- 1 Jahr Festanstellung bei einer Internet-Agentur als Lead-Entwickler, IT-Trainer und persönlicher Ansprechpartner der 15-25 Kollegen bei Problemen.
- Über 15 Jahre Selbstständigkeit als Troubleshooter, Lead-Entwickler, Software-Architekt. Hauptsächlich als Troubleshooter für IT-Teams.

Software-Entwicklung
Micha Barthel
Mühlenstrasse 136 B
28779 Bremen

E-Mail: jujumicha@web.de



Betreff: Projekte

MULTI-SERVER-ANWENDUNG¹

Optimierung und Programmierung einer Multi-Server-Anwendung zur Synchronisation von Fabrik-Logistik und -Maschinen für Autoteile. Die Maschinen standen in unterschiedlichen Ländern. Diese sollten sich automatisch nachsynchronisieren, falls mal eine einzelne Maschine offline geht.

Tätigkeit:

- Software-Architekt (Refactoring)
- Lead-Entwickler
- IT-Trainer
- Prozessoptimierung durch eigene Synchronisations-Algorithmen

Werkzeuge:

- Java
- Spring-MVC
- Spring-Boot
- Gradle
- Docker
- PostgreSQL
- RESTful-Web-Services
- Bootstrap, CSS
- jQuery & Ajax-Interface
- Jira, Confluence
- Git
- Domain Driven Design
- GitLab-CI
- Scrum

MAGENTO-SHOP MIT 60 GB DATEN

„Micha Barthel hat ohne Vorwissen ein Kundenprojekt auf Magento-Basis mit über eine Millionen Kunden optimiert und als Kernel-Entwickler anspruchsvolle Systemfehler entfernt. Die Planung und Konzeption von komplexer Softwarearchitektur ist seine Leidenschaft.“ - Alexander Boroš

Tätigkeit:

- Kernel-Entwickler
- Remote-Entwickler

Werkzeuge:

- Magento-Core
- MySql
- System-Update
- Vendor-Module
- Composer
- RESTful-Web-Services für Ki-Bot-Schnittstelle
- Trello
- Dokumentation für Ki-Bot und Server-Update

RECHTE-SYSTEM¹

Sicherheits-Optimierung und Programmierung einer Datenerfassungs-Software zur Einhaltung rechtlicher Bestimmungen großer Konzerne nach der DSGVO. Z.B. Tracking der Kunden oder Kameraüberwachung in den Supermärkten.

- Multiformular-Management mit Unterformularen
- Versionierung der Formulare
- Ausfüllhilfen für Benutzer
- Benutzergruppen und Rechtesystem
- Optimierung der Anwendung gegen Hacker

Tätigkeit:

- Software-Architekt (Refactoring)
- Lead-Entwickler

Werkzeuge:

- PHP/Laravel
- Docker
- MySql
- Jira, Confluence
- Git
- Bootstrap, CSS
- Domain Driven Design
- RESTful-Web-Services
- Scrum
- Multi-User-Rechtesystem für Benutzer, Redakteure & Admin

PIM-DATENBANK¹

Import einer PIM-Datenbank (ähnliche wie akeneo) über eine zirka 20GB großen XML-Datei. Die XML-Datei besteht aus 30-50 Datenbank-Tabellen und insgesamt 30.000 Objekten ohne Anleitung und Tabellenbeschreibung in 6 verschiedenen Sprachen. Die Datenbank ist Objektorientiert und erbt die Eigenschaften der oberen Objekte in der XML-Datei. Aus diesem Grund musste ein Parser entwickelt werden, der die Relationen in der XML-Datei abfragen kann und die Vererbung berücksichtigt. Die Vererbung wurde über einen Object-Trail auf Typo3/Extbase übertragen, so dass die Daten in einer Relationalen Datenbank abgefragt werden können über eine einzige Tabelle. Zusätzlich wurde berücksichtigt, dass es trotzdem möglich ist lesbare RealUrls zu erzeugen, welche die Abhängigkeiten der Objektdatenbank darstellen (Kategorien).

Tätigkeit:

- Software-Architekt
- Lead-Entwickler
- SAP-OBJ-Schnittstellen-Entwicklung

Werkzeuge:

- Extbase/Fluid
- TypoScript
- PHP-XML Parser
- 7Lts
- Mysql
- XML (wahrscheinlich von einer SAP-Datenbank)
- Mehrsprachiger Katalog
- Git
- Algorithmen für die rekursive Baumstruktur

BUCHHALTUNG-SOFTWARE¹

Programmieren meiner eigenen kleinen Buchhaltungssoftware mit Symfony 4.0. Entwickeln einer Symfony-Plugin Schnittstelle für Wallpage.

Tätigkeit:

- Software-Architekt
- Lead-Entwickler

Werkzeuge:

- Composer
- Doctrine / ORM/ Repository
- Twig / Form
- Console / Authentication
- PDF-Bibliotheken

SPRACH-EXPORT¹

Umstellen der Standardsprache des gesamten TYPO3-Portals. Aufbereitung des Sprach-Exports für die Übersetzungs-Agenturen für einen sicheren Import und Export über Skripte.

- Daten Import/Export für die Übersetzung durch Übersetzungsagenturen in über 15 Sprachen und jeweils über 1500 Produkt- Artikeln
- Default-Language Switch - TYPO3, umparsen von 2000 Produkt-Artikeln in 10 verschiedenen Sprachen und über 50 Datenbanktabellen
- Update von 6Lts auf 7Lts
- Entfernen diverser Core-Bugs (Gridelements)
- Optimierung mehrerer Extensions
- nginx SEO- Redirect Analyse und automatische Überprüfung

Tätigkeit:

- Software-Architekt
- Lead-Entwickler

Werkzeuge:

- Extbase/Fluid
- TypoScript
- Ext: uidcontent, gridelements, l10nmgr
- nginx
- 6Lts + 7Lts
- Mysql

DOKUMENTEN-MANAGEMENTSYSTEM FÜR KRANKENKASSE¹

Internes Dokumentenmanagementsystem für die Verwaltung der Kundendaten der bekannten KRANKENKASSE aus Berlin auf Basis von Extbase/Fluid.

- Aufbau komplexer Formulare mit bis zu 5 Unterformularen
- Solr Suchmaschinen Indexierung
- Entwicklung einer eigenen standardisierten Bibliothek der 6-8 TYPO3 Extension
- Verschlüsselung der Daten in ein externes Verzeichnis

Tätigkeit:

- Software-Architekt (Refactoring)
- Lead-Entwickler

Werkzeuge:

- Extbase/Fluid
- Formulare mit bis zu 5 Unterformularen als Obj
- Ext:Solr Hook (Bilder-Ansicht)
- Ext: FeUser, Ext: VHS
- Helper-Lib für die 6-8 Ext:
- MySql
- Git
- Jira
- Scrum

CMS¹

Entwicklung eines PHP 7 high performance CMS für performante dynamische mehrsprachige Internet-Applikationen. Unter anderem:

- Entwicklung einer Template-Engine
- Entwicklung eines Caching-Frameworks
- Flexibler Router für dynamische Applikationen
- SEO-Pagespeed Optimierung für Google PageSpeed 100/100
- Implementierung von CSS- / JS- / HTML- / IMAGE-Kompression und GZIP- ENCODING
- Entwicklung eines eigenen Plugin und Theme Management
- Frontend-Bearbeitung
- PHP 5-8 Test-Server

Tätigkeit:

- Software-Architekt
- Lead-Entwickler
- Framework-Entwicklung

Werkzeuge:

- Caching-Framework
- Template-Engine
- Flexibler Router
- Performance Optimierung
- Pagespeed Optimierung
- Theme-Builder
- SEO Optimierung

MULTI-PAY¹

Programmierung einer Multi-Gateway-Bezahlschnittstelle u.a. für die Bezahlung per Mastercard, Visa oder Paypal.

Tätigkeit:

- Lead-Entwickler

Werkzeuge:

- Extbase/Fluid
- Omnipay
- Composer
- Docke

FILM SHOP¹

Anpassen der Bezahlschnittstelle des Magento-Shops für die DHL-Informationen und den Datenexport in externes Buchhaltungssystem. Zusätzlich wurden noch 4-6 weitere Module für den Shop entwickelt.

Tätigkeit:

- Lead-Entwickler

Werkzeuge:

- Magento-Core
- Zend Framework
- Bezahlungssystem
- Datenexport in externes Buchhaltungssystem

BRILLE-SHOP¹

Programmieren eines Media-Management-System für die Verwaltung von Brillen-Bildern.

Tätigkeit:

- Lead-Entwickler

Werkzeuge:

- Magento
- Zend Framework

FLASH-PHP¹

Bugfixen einer bestehenden PHP-Applikation. Der Code bestand aus automatisch erstellten kryptischen Quellcode der von einer Flash-Anwendung erstellt wurde. Dies bedeutet, dass es keine Funktions-, Objekt-Namen mehr gab und der Quellcode komprimiert war zusammen mit den Bibliotheken.

Tätigkeit:

- Refactoring
- Lead-Entwickler

Werkzeuge:

- Magento-Core
- Zend Framework
- Bezahlungssystem
- Datenexport in externes Buchhaltungssystem

TYPO3 - OUT OF THE BOX¹

Über eine intelligente Eltern-Kind-Vererbung wird jede neue TYPO3-Webseite zu einer Distribution. Alle Abhängigkeiten werden automatisch in der richtigen Version nachgeladen. Und alle wichtigen Datenbank-Einstellungen und TS-Konfiguration werden automatisch eingestellt. Hier mein erstes Projekt, welches über einen T3N-Artikel 2008 viel Begeisterung ausgelöst hatte.

„Ich bekam über 100 E-Mails aus der ganzen Welt und auf einer Open-Source-Messe kannte man mich plötzlich. Viele Firmen erzählten mir davon, dass sie meinen Artikel gelesen hatten.“

Tätigkeit:

- Software-Architekt
- Lead-Entwickler

Werkzeuge:

- TYPO3-Core
- PHP 3, PHP 4, PHP 5
- Apache
- Ajax/ JavaScript
- Session-Management für abgebrochene Downloads
- Entwicklung eines CSS-Frameworks
- Entwicklung eines Konfigurations-Frameworks

REDAKTIONS-SYSTEM¹

Erstes TYPO3-Projekt für die Bundesregierung. Es wurden damals mehrere Backend-Module für die Datei und Artikel-Verwaltung entwickelt. Bei der Entwicklung dieser Anwendungen war ich Chef-Entwickler.

- Redaktions-System
- Media-Assat-Management-System
- Umprogrammieren des Wysiwig-Editors

Tätigkeit:

- Troubleshooter & Core-Entwickler
- Lead-Entwickler

Werkzeuge:

- TYPO3-Core
- PHP
- Bugfixing

BITV-TEST¹

Programmieren eines Parsers (Algorithmen) zur automatischen Analyse auf BITV und W3C. Dieses Programm sollte später für die Behörden der Bundesregierung eingesetzt werden.

Tätigkeit:

- Software-Architekt
- Lead-Entwickler

Werkzeuge:

- Java Swing & J2EE
- Oracle
- HTML-Parser
- PL/Sql

LANDINGPAGE-PORTAL¹

Programmieren eines Internet-Portals ähnlich wie "About.Me". Automatische Multi-Webseiten-Plattform um Landingpages und Webseiten automatisch zu generieren ohne Programmierkenntnisse mit einem Klick. Ziel war ein Internet-Portal wie "wix.com" zu entwickeln.

Tätigkeit:

- Software-Architekt
- Lead-Entwickler

Werkzeuge:

- Zend Framework 2
- MySQL
- SEO Optimierung
- Performance Optimierung
- Multi-Domain-Router

BEUTH-SHOP¹

Programmieren von PL/SQL-Skripten für die Analyse großer Datenbestände (10-20 GB). Erstellung mehrerer temporären Tabellen als Zwischenspeicher für die Sortierung. Insgesamte Ausführungszeit der Datenaufbereitung betrug über 1 Woche. Zusätzlich wurden diese Daten in der Suche eines eigens entwickelten Multi-Web-Shop verwendet. Der Web-Shop verwendet JBoss und J2ee als Grundlage und hatte mehrere Extra-Shops für verschiedene große deutsche Firmen.

Tätigkeit:

- Oracle-Entwickler
- Java-Entwickler
- Entwicklung von eigenen Sortier-
Algorithmen in PL/SQL

Werkzeuge:

- PL/Sql
- Oracle
- J2EE
- JSF
- Jboss
- SVN

WEBSHOP-PHP¹

Als Autodidakt habe ich mir in 3 Monaten das Programmieren selbst beigebracht, in dem ich ganz alleine einen kompletten Webshop mit Bezahlsystem programmiert hatte. Bereits damals hatte ich schon den Quellcode systematisch geordnet.

Tätigkeit:

- Software-Architekt
- Lead-Entwickler

Werkzeuge:

- HTML, CSS
- Javascript
- Apache, PHP, SQL

¹ Firmen-Aufträge: Die meisten meiner Arbeiten entstanden im Rahmen von Firmen-Aufträgen mit Verschwiegenheitserklärung. Diese Projekte kann ich Ihnen leider nicht öffentlich zugänglich machen. Leider habe ich erst in den letzten 2 Jahren angefangen, Referenzen zu sammeln.

Software-Entwicklung
Micha Barthel
Mühlenstrasse 136 B
28779 Bremen

E-Mail: jujumicha@web.de



Betreff: Kurzer Artikel über Monitoring, Testing & Hacking

Monitoring von Spring-Boot-Anwendungen

Das Actuator-Modul unterstützt Sie bei der Überwachung Ihrer produktiven Spring-Boot-Anwendung. Das Monitoring-Tool liefert über HTTP/JMX-Endpoints nützliche Informationen über den Zustand der laufenden Spring-Applikation.

Durch **Modultests** wie z.B. JUnit wird bei der Entwicklung die technische Lauffähigkeit und die korrekte fachliche Implementierung überprüft. Im nächsten Schritt sollen **Integrations-** und **Systemtests** sicher stellen, dass die gesamte Anwendung korrekt funktioniert. Viele Fehler werden jedoch im Programmablauf erst durch das **Monitoring** sichtbar. Besonders im Live-Betrieb ist es wichtig, Fehler in der Anwendung schnell zu lokalisieren. Über die Actuator-Endpoints kann z.B. die Verarbeitungsdauer, der Speicherverbrauch und die Anzahl der Aufrufe überprüft werden. Die Ausgabe des Actuator-Modul kann in `application.properties` oder durch selbstprogrammierte Erweiterungen angepasst werden. Externe Monitoring-Tools wie Prometheus, Zipkin, Wavefront bieten darüber hinaus hervorragende Dashboards zur übersichtlichen Darstellung von Grafiken, Analysen und Warnungen.

1. Aktivieren des Actuator-Moduls:

Durch das Hinzufügen der Abhängigkeit `spring-boot-starter-actuator` in der Maven-Datei `pom.xml` und dem Ausführen des Build-Befehls wird das Actuator-Modul aktiviert:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

In der Konsole erscheint dann die folgende Meldung:

```
Exposing 2 endpoint(s) beneath base path '/actuator'
```

2. Endpoints abfragen:

Die Endpoints sind über Http oder JMX erreichbar. Über die URL <http://localhost:8080/actuator/> erhalten wir Localhost alle verfügbaren Endpoints im JSON-Format zurück:

```
{
  "_links":{
    "self":{
      "href":"http://localhost:8080/actuator",
      "templated":false
    },
    "health":{
      "href":"http://localhost:8080/actuator/health",
      "templated":false
    },
    "health-component":{
      "href":"http://localhost:8080/actuator/health/{component}",
      "templated":true
    },
    "health-component-instance":{
      "href":"http://localhost:8080/actuator/health/{component}/{instance}",
      "templated":true
    },
    "info":{
      "href":"http://localhost:8080/actuator/info",
      "templated":false
    }
  }
}
```

3. Health-Endpoint abfragen:

Über die URL <http://localhost:8080/actuator/health> wird uns angezeigt, ob die Anwendung fehlerfrei läuft. Bei UP ist alles ok. Bei DOWN haben wir z.B. Probleme mit der Konnektivität der Datenbank oder zu wenig Speicherplatz.

4. Info-Endpoint abfragen:

Über die URL <http://localhost:8080/actuator/info> bekommen wir zunächst ein leeres JSON zurück. Die Informationen über die Anwendung müssen wir erst über `application.properties` zur Verfügung stellen:

```
# INFO ENDPOINT CONFIGURATION
info.app.name=@project.name@
info.app.description=@project.description@
info.app.version=@project.version@
info.app.encoding=@project.build.sourceEncoding@
info.app.java.version=@java.version@
```

5. Weitere Endpoints aktivieren:

Aus Sicherheitsgründen sind weitere Actuator-Endpoints über HTTP deaktiviert. Nur über JMX können alle Endpoints direkt abgefragt werden. Die vollständige Liste finden Sie unter:

<https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-features.html>

Den jeweiligen Endpoint aktivieren wir in den `application.properties` durch:

```
management.endpoint.<Actuator-Endpoint-Id>.enabled = true
```

Wir nehmen als Beispiel den Endpoint **beans**:

```
management.endpoint.beans.enabled = true
```

Nun müssen wir in den `application.properties` noch angeben, welche Endpoints veröffentlicht werden sollen. **info** und **health** müssen ebenfalls eingetragen werden.

```
management.endpoints.web.exposure.include=info, health, beans
```

6. Eigenen Endpoint programmieren:

Ein Endpoint (z.B. `/actuator/custom`) kann mit einer Bean erstellt werden:

```
import org.springframework.boot.actuate.endpoint.annotation.Endpoint;
import org.springframework.boot.actuate.endpoint.annotation.ReadOperation;
import org.springframework.context.annotation.Bean;
import org.springframework.stereotype.Component;

no usages
@Endpoint(id="custom")
@Component
public class CustomActuatorEndpoint {
    no usages
    @ReadOperation
    @Bean
    public String getCustomActuatorEndpoint() {
        return "/actuator/custom liefert eigene Informationen über das System!";
    }
}
```

Der neue Endpoint muss in den `application.properties` aktiviert werden:

```
management.endpoint.custom.enabled = true
```

Jetzt müssen wir den Endpoint in den `application.properties` noch veröffentlichen:

```
management.endpoints.web.exposure.include=info, health, beans, custom
```

7. Endpoints vor Hackern schützen:

Hacker erhalten sehr viele Systeminformationen über die Actuator-Endpoints, wenn das Security-Modul nicht richtig eingestellt ist. Angreifer können z. B. die `reloadByUrl`-Funktion und den JNDI-LDAP-Hack verwenden, um einen Exploit auszuführen:

<https://www.veracode.com/blog/research/exploiting-spring-boot-actuators>

Durch das Hinzufügen der Abhängigkeit `spring-boot-starter-security` in der Maven-Datei `pom.xml` und dem Ausführen des Build-Befehls wird das Security-Modul aktiviert:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

In den `application.properties` aktivieren wir jetzt einen Admin. Damit sollte der Zugriff auf die Endpoints nur noch über ein Login möglich sein:

```
spring.security.user.name=Admin
spring.security.user.password=78fa095d-3f4c-ad50-e24c31d5cf35
spring.security.user.roles=ADMIN
```

Jetzt müssen wir noch eine `SecurityConfiguration`-Bean anlegen, damit der Actuator-Endpoint `http://localhost:8080/actuator/*` nur noch für einen User mit der Rolle `ADMIN` erreichbar ist:

```

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;

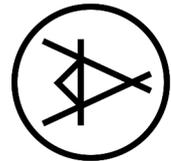
no usages
@Configuration
@EnableWebSecurity
public class WebSecurity {
    no usages
    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http.authorizeHttpRequests()
            .requestMatchers("/actuator/**").hasRole("ADMIN")
            .anyRequest().authenticated()
            .and()
            .httpBasic();
        return http.build();
    }
}

```

Fazit

In diesem kurzen Leitfaden habe ich zu erklären versucht, warum die Überwachung bzw. Protokollierung für die Überprüfung der Funktionsfähigkeit der Anwendung äußerst wichtig ist. Zu diesem Zweck habe ich eine kurze Einführung in die Verwendung des Actuator-Moduls geschrieben. **UnitTests** liefern zu wenig Informationen über die Funktionsfähigkeit der kompletten Anwendung, da hier nur die Qualität der einzelnen Einheiten überprüft wird und nicht die Qualität des kompletten Systems. Ohne reale Daten aus dem Live-Betrieb können **Integrations-** und **Systemtests** ebenfalls nur eingeschränkt die Funktionsfähigkeit der Anwendung überprüfen. Die Qualität einer Anwendung kann nur gewährleistet werden, wenn regelmäßig Kontrollen von verschiedenen Personen durchgeführt werden. Das Thema **Sicherheit** wird bei der Entwicklung oft vernachlässigt, da der Fokus oft mehr auf die Umsetzung der Funktionalität liegt. So kann es leicht passieren, dass bei einem Release vergessen wird, die Anwendung bestmöglich gegen Hacker abzusichern. Natürlich wird ein Angreifer nicht den individuellen Quellcode der Spring-Boot-Anwendung kennen. Es reicht jedoch aus, einen gängigen Tomcat- oder Spring-Exploit zu verwenden, um Zugriff auf das gesamte System zu erhalten. Daher ist es wichtig, dass Updates immer sofort von einem professionellen DevOps-Engineer eingespielt werden.

Software-Entwicklung
Micha Barthel
Mühlenstrasse 136 B
28779 Bremen



E-Mail: jujumicha@web.de

Betreff: Artikel über eine selbst entwickelte agile Methode

Die Methode SXX

SXX ist ein gesunder Mix für die professionelle Software-Entwicklung im Team. SXX beseitigt lustlose und damit sinnlose Daily-Scrums, nicht funktionierendes Pair-Programming und stellt unterschiedliche Talente jedes Team-Mitglieds in den Vordergrund. SXX erweitert Scrum, indem es mit Kanban den Workflow optimiert und mit XP wichtige Praktiken der Software-Entwicklung einführt.

In der folgenden Beschreibung werden nur die Unterschiede von SXX zu Scrum erläutert, da SXX auf Scrum aufbaut und Methoden von XP und Kanban zusätzlich benutzt. Es wird nur auf die pragmatische Anwendung eingegangen und nicht auf ein sogenanntes Mindset. SXX hat das Ziel, dass sich das Mindset durch die Anwendung automatisch ergibt und man sich dieses nicht erst gläubig einbilden muss, wie bei den anderen Agilen Methoden. SXX baut Zusammenhänge zwischen den verschiedenen Methoden Scrum, XP und Kanban auf und gibt den Scrum-Regeln einen Sinn. Workflow und Techniken muss man neben diesen Regeln ständig optimieren und anpassen.

Fremdwörter? Kurz und knapp!

Agile Methoden zeichnen sich durch selbstorganisierende Teams aus. Dies führt zu mehr Selbstvertrauen im eigenen Handeln. Dadurch kann man zusammen flexibel auf veränderte Anforderungen des Kunden reagieren. Offene Kommunikation und Wissensaustausch stehen im Vordergrund. Es wird immer nur das Wesentliche für ein funktionierendes Produkt umgesetzt.

Scrum ist eine agile Methode für das Projektmanagement. Ein Projekt wird immer nur für 2-4 Wochen schrittweise zusammen mit dem Kunden geplant und in diesem Zeitraum in kleineren Aufgaben abgearbeitet und fertig gestellt. Danach wird der nächste Schritt geplant, abgearbeitet und fertig gestellt.

XP oder Extreme Programming funktioniert ähnlich wie Scrum. XP macht allerdings enge Vorgaben für die Qualitätssicherung professioneller Software-Entwicklung.

Kanban ist eigentlich keine agile Methode und dient hauptsächlich zur Optimierung des Workflows der Arbeitsschritte. Aus diesem Grund findet Kanban auch Anwendung bei nicht projektbezogener Arbeit.

SXK Techniken

Im Folgenden werden SXK-Techniken und Abweichungen zu Scrum erklärt:

A. SXK-Workflow: Arbeitsschritte des Taskboards

Durch Kanban wird der Workflow in seine wichtigsten Schritte unterteilt. Kanban führt für bestimmte Schritte WIP "Work in Progress" ein. Dies limitiert die Anzahl der Tickets, die in einem Schritt bearbeitet werden dürfen. Die Anzahl ist flexibel und soll ein Aufstauen von Tickets verhindern. Mehrere Tickets können, wenn es Sinn macht, zu einem Epic zusammengefasst werden.

1. **Product-Backlogs**
2. **SPRINT: ToDo**
 1. Sprint-Backlog
 2. Return-Log: Manchmal müssen halb fertige Tickets wieder zurückgestellt werden.
3. **SPRINT: In Progress**
 1. Architektur [WIP: onGoing/Done]
 2. Unit-Test [WIP: onGoing/Done]
 3. Development [WIP: onGoing/Done]
 4. Code-Review & Refactoring [WIP: onGoing/Done]
 5. Test [WIP: onGoing/Done]
4. **SPRINT: Done**
 1. Done
 2. Inkrement Done: Hier kommen nur die Tickets hinein, die auch wirklich in das Inkrement kommen.
5. **Fertige "Product Incremente"**

B. Aufbau eines Tickets:

1. User-Story (Mehrere User-Stories können auch als Epic-Ticket zusammengefasst werden.)
2. Kompetenz-Gebiet
3. Art (Bug, Feature)
4. Min/Max Zeit
5. Zeit gebraucht
6. Prio
7. Status (ToDo, InProgress, Done)

C. Kompetenz-Gebiete im SXX:

Im SXX ist es wichtig, dass jedes Team-Mitglied seine individuellen Talente und Kompetenz-Gebiete kennt. Ein einzelnes Kompetenz-Gebiet wird immer von 2 Team-Mitgliedern gepflegt, damit im Krankheitsfall jederzeit einer einspringen kann. Ein SXX-Team ist stark, wenn es viele unterschiedliche Persönlichkeiten mit verschiedenen Talenten gibt.

D. Wissensaustausch und Pair-Programming im SXX:

Pair-Programming ist eine XP-Arbeitstechnik, bei der 2 Entwickler immer an einem Computer arbeiten. Dies soll den Wissensaustausch im Team verbessern. Wenn der eine Entwickler eine lernende Rolle und der andere Entwickler eine lehrende Rolle einnimmt, kann dies Sinn ergeben. Viele Entwickler mögen allerdings das Pair-Programming nicht, weil sie dadurch weniger selbst entscheiden können und sich womöglich kontrolliert fühlen. Im SXX gibt es Kompetenz-Gebiete. Hierbei wird das Projekt in Kompetenz-Gebiete unterteilt. Da 2 Team-Mitglieder ein Kompetenz-Gebiet pflegen, wird ein einzelnes, kleines Aufgaben-Ticket im Kanban-Tandem-Workflow bearbeitet. So erstellt der erste Entwickler die Architektur oder den Unit-Test für die Aufgabe oder macht sich grob Gedanken darüber, worum es in der Aufgabe geht und schreibt diese mit seinem technischen Wissen ins Aufgaben-Ticket. Jetzt programmiert der zweite Entwickler diese Aufgabe nach seiner Vorstellung mit dem Wissen des Kompetenz-Partners fertig (...währenddessen programmiert der erste Entwickler an dem vorbereiteten Ticket des ersten Entwicklers!). Im letzten Schritt macht der erste Entwickler eine technische Qualitäts-Kontrolle. Beide Entwickler bearbeiten gleichzeitig immer alle Bearbeitungs-Schritte im Kanban-Tandem-Workflow, mit dem Unterschied, dass bei einer Aufgabe die Schritte abwechselnd zwischen den beiden Entwicklern hin und her geschoben werden. Dadurch tauschen beide Entwickler ständig Wissen aus und verbessern so die soziale Kommunikation.

E. Das perfekte Entwickler-Team:

Jeder Mensch ist unterschiedlich und kein Entwickler kann alles. Auch ein Entwickler, der angeblich alles kann, ist in seiner Kompetenz begrenzt. Ein solcher Entwickler muss lernen, eine kontrollierende Rolle im Team einzunehmen, indem er Software-Architekt wird, eine vorausschauende, planende und kontrollierende Tätigkeit einnimmt oder andere Entwickler anleitet. Es werden auch immer mehr technische Product-Owner gesucht. Also Hardcore-Experten, die nur noch die Planung machen. Es gibt allerdings auch sehr fokussierte Nerds, die sich mit großer Leidenschaft auf die größten Probleme stürzen und keinen großen Wert auf den kommunikativen Austausch

legen. Diese sollten sich besser selbstständig auf die Probleme konzentrieren. Als letztes haben wir die begeisterten Bastler. Ohne diese kann kein Projekt fertig gestellt werden. Auf diesem Gebiet hat der Experte oft seine Schwachstellen, da manchmal die Umsetzung wichtiger ist als ständiger Perfektionismus. Bei SXK muss nicht jeder Entwickler auch ein perfekter Software-Entwickler sein. Ein perfektes Entwickler-Team haben wir nur, wenn wir viele unterschiedliche Persönlichkeiten und Talente haben.

F. Kreative Ideen:

Ein innovatives Team tauscht Wissen aus und spricht offen über Hindernisse im Projekt. Diese Ideen und Probleme werden auf einem öffentlichen (nicht digitalen) Ideen-Board auf einem Notizzettel festgehalten. Jedes Team-Mitglied hat ein eigenes Ideen-Board, so dass man beim nächsten Daily-Scrum sofort sehen kann, worüber man in den nächsten 15 Minuten reden könnte. Unausgesprochene Ideen werden ganz links festgehalten. In der zweiten Spalte werden die besprochenen Ideen festgehalten. In der dritten Spalte werden die "krummen" und vielleicht irrationalen Ideen manifestiert. In der letzten Spalte werden die Super-Ideen festgenagelt.

G. Planung im SXK:

Im SXK ist das Planen durch den Product-Owner eine wichtige Aufgabe. Natürlich ist inzwischen allen in der Software-Entwicklung klar, dass es schwer ist, eine technische Spezifikation oder Kunden-Spezifikation im Voraus zu erstellen, die wirklich eingehalten wird. Allerdings spricht nichts dagegen, User-Stories oder eine Spezifikation im Voraus mit dem Kunden zu erstellen. Diese dient als roter Pfaden, damit das Projekt nicht vom Weg abkommt. Eine Spezifikation dient als Wissensgrundlage, da man in der agilen Software-Entwicklung in 2-4 wöchigen Schritten (Sprints) die Software plant und umsetzt. Ein Sprint hat immer 50% "Feature" und 50% "Bug+Nice to have". Dies soll sicherstellen, dass in einem Sprint immer alle "Features" erfolgreich abgeschlossen werden. Das Team wird sich freuen, wenn es frühzeitig alle Ziele erreicht hat und kann sich dann mehr um die Qualität der Software kümmern und zusätzliche sinnvolle Features implementieren. Ziel ist es, dass das Team stressfrei 100% Leistung erreicht durch Anspannung und Entspannung.

Grundlagen aus dem Scrum

Scrum ist eigentlich ganz einfach. Natürlich kann man sich ganz viele Bücher dazu durchlesen, aber man sollte neben diesen Regeln und Schritten Scrum ganz pragmatisch angehen und einfach anwenden. Zusätzlich sollte man sich noch mal das Agile Manifest zu Gemüte führen. Ein Team kann nur stark sein, wenn man Mitmenschen und Kunden respektiert.

Die 3 Rollen eines Teams

Damit ein Projekt überhaupt in Angriff genommen werden kann, muss dieses vom Product-Owner, dem Planer, geplant werden. Dieser ist gleichzeitig Fach-Experte und hat das Ziel, die Kundenwünsche zu erfüllen. Für das Einhalten der agilen Regeln ist der Kommunikator zuständig. Oft wird dieser auch Scrum-Master genannt. Dieser kümmert sich um das Team und ist kein Projektleiter. Das Team darf selbstständig agil entscheiden und erfüllt somit zusammen die Rolle eines Projektleiters. Das Team selbst ist für die Umsetzung des Produktes zuständig. Eine Rolle darf nicht doppelt besetzt werden.

Workflow

- Product Owner plant und sortiert den Product-Backlog nach Priorität
- SPRINT von 1-4 Wochen
 1. MEETING: Sprint-Planning: Team entscheidet, welche Aufgaben (User-Stories) in den Sprint-Backlog zum Abarbeiten kommen.
 2. TASKBOARD: Aufgaben (User-Stories) werden vom Entwickler-Team einzeln in den Schritten "ToDo", "InProgress" und "Done" abgearbeitet.
 3. Täglich wird ein DAILY-Scrum von 15 Minuten gemacht.
 4. Wenn das SPRINT-Inkrement "Teil-Produkt" fertig ist, trifft sich das Team mit dem Kunden und Stakeholder zu einem Sprint-Review.
 5. MEETING: Danach trifft sich das komplette Scrum-Team zu einer Sprint-Retrospektive und entscheidet selbst, was man beim nächsten Sprint besser machen könnte.
- Das Produkt ist fertig, wenn alle SPRINT-Inkremente umgesetzt wurden. Es wird immer nur im nächsten Sprint entschieden, welche Aufgaben (User-Stories) umgesetzt werden.

Software-Entwicklung
Micha Barthel
Mühlenstrasse 136 B
28779 Bremen

E-Mail: jujumicha@web.de



Betreff: Eigener Artikel aus der T3N

<https://t3n.de/magazin/typo3-out-of-the-box-t3s-installationsprogramm-220539/> (29.02.2008)

T3S: Ein Installationsprogramm für TYPO3-Distributionen

Das Team um T3Sugar hat sich zum Ziel gesetzt, die Installation eines TYPO3-Auftritts für den Anwender so einfach wie möglich zu gestalten. Die Idee hinter der Out-of-the-Box-Installation für TYPO3: Mit einem Installations-programm für frei verfügbare T3S-Distributionen können Anwender einen kompletten Webauftritt innerhalb weniger Minuten aufsetzen.

Eine neu aufgesetzte TYPO3-Installation ohne Erweiterungen verfügt nur über wenige Eigenschaften. Man kann damit eine typische Web-1.0-Website mit statischem Text und Navigation darstellen. Interessant wird TYPO3 erst, wenn erschienen in man sich der vielen Erweiterungen bedient. Prinzipiell deckt TYPO3 mit Hilfe der vorhandenen Erweiterungen die meisten Anwendungsbereiche im Internet ab. Die Installation der Erweiterungen gestaltet sich jedoch im Gegensatz zu dem Aufsetzen der Grund-installation nicht immer leicht und ist sehr zeitaufwändig.

Einsatzbereiche für eine T3S-Distribution:

Business	Forum	Wiki
Blog	Shop	Homepage
Event	Portal	Community

Fertig verpackt als T3S-Distribution wird die Installation eines fix und fertigen TYPO3-Auftritts jedoch zu einem Kinderspiel. Ein flexibles Baukasten-System erleichtert die Anpassung der neu aufgesetzten Distribution und macht das Importieren und Exportieren einer selbst erstellten Distribution möglich. Die T3S-Distribution ist eine Art Konfigurationspaket und fügt die Software von

TYPO3 zu einem sinnvollen Ganzen zusammen.

Eigenschaften der Standard-T3S-Distribution:

Eigenschaft	TYPO3-Erweiterung (Extension)
SEO	RealURL, CoolURI
Medienverwaltung	DAM
Mehrsprachigkeit	sb_languageswitcher
Nachrichten-Teaser	tt_news
Template-Umschalter	rlmp_tmplselector, templavoila
CSS-Framework	T3sugarCSS
Extension Entwicklung	lib, div, kickstarter
Administration	phpmyadmin, ter_update_check

Die eigene T3S-Distribution

Bei jeder Installation mit dem T3S-Install-Tool wird automatisch eine eigene T3S-Distribution für einen bestimmten Einsatzbereich erstellt. Die Distribution ist nach der Installation nichts anderes als eine Kopie der Distribution, die man bei der Installation mit dem T3S-Install-Tool ausgewählt hat und wird für den neu erstellten TYPO3-Auftritt als Konfigurationspaket installiert. Die Installations-Distribution, die man für die Einrichtung mit dem T3S-Install-Tool anwendet, wird danach nicht mehr benötigt.

Die eigene T3S-Distribution kann man nach eigenen Bedürfnissen anpassen und später in Form einer TYPO3-Erweiterung (Extension) jedem zugänglich machen. So wird die Wiederverwendbarkeit eines fertig konfigurierten TYPO3-Auftritts ermöglicht.

T3S-Installation:

1.	Neue TYPO3-Installation aufsetzen (neue Datenbank, ohne Erweiterungen) mit dem Dummy-Paket von TYPO3
2.	Extension T3S über den Extension-Manager installieren (keine weiteren Extensions!)
3.	Komplettes TYPO3-Backend neu laden über die Refresh-Taste des Browsers
4.	Das T3S-Install-Tool ausführen, das unter Tool im Backend-Menü erscheint
5.	Bei der Installation alle Hinweise über den jeweiligen Link verfolgen
6.	Cache löschen (T3S-Distribution ist eingerichtet!)

Baukasten-System

Nach der Installation von T3S steht dem Anwender für seine eigene T3S-Distribution der T3S-Editor zur Verfügung. Das ist ein in das Redaktionssystem integriertes Backend-Modul mit mehreren Untermodulen. Dieser T3S-Editor gliedert die verschiedenen Konfigurationsbereiche von TYPO3. Die Konfiguration von TypoScript und TSconfig wird systematisch nach den Eigenschaften in einzelne Ordner eines Verzeichnissystems aufgeteilt. Die Konfiguration des Rich-Text-Editors (RTE) lässt sich so zum Beispiel leichter

wiederfinden. Darüber hinaus hat der RTE dadurch das gleiche Layout wie auf der erzeugten Website.

Auch für das Layout wird ein eigenständiges Modul mitgeliefert. Der Webdesigner modifiziert das bestehende Layout aus Templates, CSS und Bildern nach den eigenen Bedürfnissen und schafft so ein funktionstüchtiges Layout für den TYPO3-Auftritt.

Das Konzept wird durch ein Setup-Modul im T3S-Editor abgerundet. Dort werden alle Aktionen festgelegt, die beim Ausführen der Installation der TYPO3-Distribution mit dem T3S-Install-Tool anfallen. Die Definition einer Liste aller herunterzuladenden Erweiterungen für TYPO3 aus dem Extension-Repository ist beispielsweise eine der anfallenden Installationsaktionen. Der T3S-Editor verwendet den Dateimanager „Quixplorer“ und integriert den Editor „T3Editor“ für die Syntaxhervorhebung der verschiedenen Dateiformate. TypoScript lässt sich so leichter bearbeiten und als Extra-Feature gibt es nach dem Speichern Informationen über die Korrektheit der Syntax unter dem Editor.

Was bietet T3S?	Wie?
Übersichtliche FE- und BE-Konfiguration	integrierter Konfigurations-Editor
TYPO3-Setup-Konfiguration	TYPO3-Setup, Extension-Setup und Server-Umgebungs-Setup
Datenbank-Vorkonfigurierung	Benutzerrechte, Seitenbaum, Beispiel-Daten
Extension-Installation	aus dem TER mit Versionsangabe
Vorkonfigurierung des T3-Dateisystems	DateisystemsFileadmin, Typo3temp
CSS-Editor-Framework	Leichtes handhaben eines CSS-Frameworks
Update-Möglichkeit	alle in der Distribution mitgelieferten Extensions
Eigene T3S-Distribution	wird automatisch erzeugt

Langfristig ist es das Ziel, TYPO3 durch T3Sugar für alle Anwendergruppen benutzerfreundlicher zu machen. Dadurch bekommt das CMS ein breiteres Einsatzspektrum und die Open-Source-Community wird weiter gestärkt.

T3S hat bereits ein stabiles Konzept zur Installation und Erzeugung von T3S-Distributionen. Jetzt fehlen lediglich noch Anwender, die eine T3S-Distribution bei sich installieren und ihre eigene neue Distribution zum Download anbieten. Sollten bei der Installation Probleme auftauchen, weil beispielsweise die Server-Umgebung von TYPO3 nicht optimal eingerichtet ist, leistet die stetig wachsende T3Sugar-Community [1] schnelle Abhilfe. Zusätzliche Informationen liefert die mehrsprachige Dokumentation [2] auf der T3Sugar-Internetpräsenz [3]. Der Extension-Autor und Projektleiter freut sich über jede

Art von Feedback und Mitarbeit. Besonders willkommen sind neue Anregungen für eine noch bessere Installation „out-of-the-box“.

Autor: Micha Barthel interessiert sich im Moment besonders dafür, wie man als Software-Architekt ein Projekt erfolgreich plant, damit jeder seine eigenen Fähigkeiten im Team voll ausnutzen kann. Seit neuestem versucht er Workshops mit T3S zu leiten. Erfahrungen gesammelt hat er im Bereich Applikationsserver mit **J2EE** und den verschiedenen **PHP-Frameworks**.