Entwicklungsplan

BIKS



Neutorstraße 13  
5020 Salzburg

AUSTRIA

Tel: +43 (662) 276198-11

Fax: +43 (662) 276198-98

Inhaltsverzeichnis

[Änderungsverzeichnis 2](#_Toc500755971)

[1 Vorwort (Foreword) 3](#_Toc500755972)

[1.1 Ziel (Purpose) 3](#_Toc500755973)

[1.2 Geltungsbereich (Scope) 3](#_Toc500755974)

[1.3 Definitionen, Akronyme, Abkürzungen (Definitions, Acronyms, Abbreviations) 3](#_Toc500755975)

[1.4 Verweise (References) 3](#_Toc500755976)

[2. Projektübersicht (Project Overview) 4](#_Toc500755977)

[2.1 Projektziele, Annahmen, Einschränkungen (Project Purpose, Assumptions, Constraints) 4](#_Toc500755978)

[3. Projektorganisation (Project Organization) 4](#_Toc500755979)

[3.1 Organisationsstruktur (Organizational Structure) 4](#_Toc500755980)

[3.2 Externe Schnittstellen (External Interfaces) 4](#_Toc500755981)

[3.3 Rollen und Verantwortlichkeiten (Roles and Responsibilities) 4](#_Toc500755982)

[„Rational Unified Prozess“-Rolle 4](#_Toc500755983)

[Projektleiter 4](#_Toc500755984)

[4. Managementprozess (Management Process) 6](#_Toc500755985)

[4.1 Prozessabschätzungen (Project Estimates) 6](#_Toc500755986)

[4.2 Projektplan (Project Plan) 6](#_Toc500755987)

[4.2.1 Phasen-Plan (Phase Plan) 6](#_Toc500755988)

[4.2.2. Iterationszyklen (Iteration Plans) 6](#_Toc500755989)

[4.2.3 Releases 6](#_Toc500755990)

[4.2.4 Projektzeitplan (Project Timetable) 6](#_Toc500755991)

[4.2.5 Projekt-Ressourcen (Project Resources) 6](#_Toc500755992)

[5. Technischer Prozess (Technical Process) 6](#_Toc500755993)

[5.1 Tools 6](#_Toc500755994)

[5.2 Richtlinien (Guidelines) 6](#_Toc500755995)

[6. Anhang (Appendix 6](#_Toc500755996)

[6.1 Personalkosten (Personal Costs) 6](#_Toc500755997)

# Änderungsverzeichnis

|  |  |  |
| --- | --- | --- |
| Datum | Version | Beschreibung |
| 2017-12-11 | 0.1 | Erstversion |

# Vorwort (Foreword)

## 1.1 Ziel (Purpose)

BIKS steht für Breanos Industry Knowledge System und steht als Sammelbegriff für eine Softwarearchitektur, die Industrie 4.0 real umsetzt. Dieses Dokument dient zur Steuerung, Kontrolle und Dokumentation des Projektfortschritts.

Die Verwaltung dieses Dokuments geschieht durch den Projektverantwortlichen, steht aber jedem Team-Mitglied zur freien Verfügung.

Die sich ergebenden Aktivitäten werden in eine schlüssige Reihenfolge gebracht um den Projektablauf detailliert festzuhalten. Die sich daraus ergebenen Kontrolltermine und Meilensteine helfen dabei, das Einhalten des Fertigstellungszeitpunktes zu kontrollieren.

Der Software-Entwicklungsplan wird von dem Projektmanager verwendet, um Planungen bezüglich Zeitplans und den benötigten Ressourcen zu treffen. Außerdem ermöglicht er die Überprüfung des Ablaufs und aktuellen Fortschritts anhand des festgelegten Zeitplans.

Die Projektmitglieder wissen dank dem Software-Entwicklungsplan welche Aufgaben sie wann zu erledigen haben und welche Abhängigkeiten zwischen den einzelnen Aktivitäten bestehen. Die Projektmitglieder sind dazu verpflichtet, sich an den Software-Entwicklungsplan zu halten und potentielle Änderungen und Abweichungen vorher mit dem Projektleiter und dem Team zu besprechen.

## 1.2 Geltungsbereich (Scope)

Dieser Software-Entwicklungsplan beschreibt die Gesamtplanung, auf deren Basis das Projektteam die Entwicklung des BIKS Projekts realisiert. Die Details der einzelnen Iterationen werden in den Iteration Plans beschrieben. Die Planungen in diesem Dokument basieren auf den Software-Anforderungen, die durch das „Software Requirement Specification“-Dokument definiert werden.

## 1.3 Definitionen, Akronyme, Abkürzungen (Definitions, Acronyms, Abbreviations)

Siehe Glossar.

## 1.4 Verweise (References)

Folgende Referenz-Dokumente stehen in Abhängigkeit mit dem Software-Entwicklungsplan:

<http://bre-tfs02:8080/tfs/BIKS%20Framework%20Collection/PM%20BIKS/_backlogs>

# 2. Projektübersicht (Project Overview)

## 2.1 Projektziele, Annahmen, Einschränkungen (Project Purpose, Assumptions, Constraints)

Das BIKS Projekt dient der Entwicklung eines funktionsfähigen Architekturprototypen für die Kernkomponenten des BIKS. Das Endprodukt soll als Grundlage für weitere Software dienen. Desweiteren wird das Architekturkonzept bewiesen, ein Testsystem aufgebaut und eine präsentierbare Demoversion implementiert.

Im Rahmen der Entwicklungsarbeit wird eine vollständige UML-Dokumentation der gesamten Architektur erstellt. Für dieses Projekt wird das neue Vorgehensmodell für die Breanos Softwareentwicklung verwendet und mit gewonnenen Erkenntnissen verfeinert.

# 3. Projektorganisation (Project Organization)

## 3.1 Organisationsstruktur (Organizational Structure)

Das Projekt wird von einem vierköpfigen Projektteam durchgeführt, je nach Verfügbarkeit.

Projektverantwortlicher: ABE

Product Owner: ABE

Scrum Master: ABE

Entwickler: EBE, GED, FKE, ABE

## 3.2 Externe Schnittstellen (External Interfaces)

Das Projektteam wird mit folgenden externen Beteiligten in Verbindung stehen:

-

## 3.3 Rollen und Verantwortlichkeiten (Roles and Responsibilities)

### Projektleiter

Achim Bernhard

#### Stellvertretender Projektleiter

Gerhard Eder

#### Scrum Master

Achim Bernhard

#### Product Owner

Gerhard Eder

#### Product Owner

Achim Bernhard

#### Software-Entwickler

Franz Kellner

#### Software-Entwickler

Eduard Bezdedeanu

#### Software-Entwickler

Achim Bernhard

#### Software-Entwickler

Gerhard Eder

Folgende Tabelle definiert die primären Verantwortlichkeiten der einzelnen Rollen:

#### Projektleiter

Verantwortlich für die operative Leitung des Projekts.   
Dies umfasst die Koordination, Steuerung, Kontrolle und Planung.

Aufgaben und Kompetenzen:

- Projektplanung

- Überwachen des Zeitplans und Budgets

- Steuerung des Projektablaufs

- Verantwortung für das Projektergebnis

- Koordination des Projektteams

#### Stellvertretender Projektleiter

Vertretung der Projektleitung. Übernimmt die Aufgaben des Projektleiters, sollte dieser verhindert sein.

#### Software-Entwickler

Verantwortlich für die Erstellung und Beschreibung der zugrunde liegenden Softwarearchitektur. Führt unterstützende Aufgaben bei der Umsetzung der Vision aus.

Aufgaben und Kompetenzen:

- Gestaltung der Software-Architektur

- Entwurf von UML-Diagrammen

- Umsetzung von textuellen Anforderungen in Modelle

- Unterstützung der Software-Designern

- Überwachung des Gestaltungsprozesses

Verantwortlich für das Design der Software und

Festlegung der Spezifikationen.

Aufgaben und Kompetenzen

- Erstellen des visuellen Erscheinungsbilds mit Hilfe von

Mockup-Tools

- Festlegung der zu verwendenden Programmiertechniken

Implementierung der vorgesehenen Funktionalitäten.

Aufgaben und Kompetenzen:

- Erstellung des Quellcodes auf Basis der entworfenen UML-Diagrammen

- Umsetzung mit objektorientierten Programmiersprachen

- Realisierung der Datenbankanbindung

- Entwicklung des Machbarkeits-Protoypen, einer Beta-Version und der finalen Version

#### Product Owner

Vertritt die Produktvision und priorisiert Tätigkeiten und Features.

Aufgaben und Kompetenzen:

- Erstellung von Work Items und User Stories

- Priorisierung von Items nach Geschäftswert

- Erklären der Produktvision und Beantworten von Fragen der Entwickler zu Features

#### Scrum Master

Sorgt für die Einhaltung der Scrum-Richtlinien.

Aufgaben und Kompetenzen:

- Erstellung von Work Items und User Stories

- Assistierende Priorisierung von Items nach Geschäftswert

- Hilfestellung bei Fragen oder Problemen mit Scrum

- Administration des TFS Scrum Portals

- Anlegen und Abschließen von Sprints

- Kontrolle abgeschlossener Arbeiten

# 4. Managementprozess (Management Process)

## 4.1 Prozessabschätzungen (Project Estimates)

## 4.2 Projektplan (Project Plan)

### 4.2.1 Phasen-Plan (Phase Plan)

### 4.2.2. Iterationszyklen (Iteration Plans)

#### Sprint 1

*Zeitraum*

22.01.2018 - 29.01.2018

*Scope*

Service Bus Kommunikation Activity

Architekturkonzeption (Maschinenschicht, KPUs, Maschinenblackboards, zentrales Blackboard)

Werkstück Grobkonzeption

*Ergebnis*

#### Sprint 2

*Zeitraum*

29.01.2018 - 05.02.2018

*Scope*

Units Konzeption, Implementierung

Blackboard Konzeption

*Ergebnis*

#### Sprint 3

*Zeitraum*

05.02.2018 - 12.02.2018

*Scope*

TCP/IP Socket Kommunikation Activity

Units Konzeption, Implementierung

Blackboard Implementierung

Fabrik mit Activities simulieren inkl Kommunikationstests

*Ergebnis*

#### Sprint 4

*Zeitraum*

12.02.2018 - 19.02.2018

*Scope*

Round-Trip von Fabrik über Blackboard zurück zu Fabrik simulieren

*Ergebnis*

#### Sprint 5

*Zeitraum*

*Scope*

UI Konzept auf Basis der simulierten Fabrik

*Ergebnis*

### 4.2.3 Releases

### 4.2.4 Projektzeitplan (Project Timetable)

### 4.2.5 Projekt-Ressourcen (Project Resources)

# 5. Technischer Prozess (Technical Process)

## 5.1 Tools

## 5.2 Richtlinien (Guidelines)

## 5.3 Durchführung

### Sprint

Sprints starten am Montag und enden am Montag der darauffolgenden Woche

Dauer: 1 Woche

### Sprint Planning

Sprint Planning beginnt am Montag nach dem Sprint Review des vorhergehenden Sprints. Dabei wird der neue Sprint angelegt und die Arbeit für den Sprint geplant und besprochen.

Dauer: 30-120 Minuten

### Daily Scrum

Jeden Tag gibt es ein Daily Scrum, wo aktuelle Herausforderungen und der Status besprochen werden.

Dauer: 15-30 Minuten

#### Sprint Review

Das Sprint Review wird am Montag vor dem Start des nächsten Sprints durchgeführt. Erledigte Items werden entfernt und die Performance des Teams dem Zeitplan gegenübergestellt.

Dauer: 15-90 Minuten

#### Sprint Retrospective

Die Sprint Retrospective ist optional und dient der Prozessanpassung und -optimierung.

Dauer: 15-120 Minuten

# 6. Anhang (Appendix)

## 6.1 Personalkosten (Personal Costs)