

Requirements and Design Documentation

(RDD)

Version 0.0

ESEP - Praktikum - Wintersemester 2016

Lüdemann	Mona	xxxxxxx	mona.luedemann@haw-hamburg.de
Butkerei	Marvin	xxxxxxx	marvin.butkerei@haw-hamburg.de
Schumacher	Wilhelm	xxxxxxx	wilhelm.schumacher@haw-hamburg.de
Melkonyan	Anushavan	xxxxxxx	anushavan.melkonyan@haw-hamburg.de
Colbow	Marco	xxxxxxx	marco.colbow@haw-hamburg.de
Cakir	Mehmet	2195657	mehmet.cakir@haw-hamburg.de

18. Oktober 2016

Änderungshistorie:

Version	Author	Datum	Anmerkungen/Änderungen

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Teamorganisation	3
1.1 Verantwortlichkeiten	3
1.2 Absprachen	3
1.3 Repository-Konzept	3
2 Projektmanagement	3
2.1 Prozess	3
2.2 PSP/Zeitplan/Tracking	3
2.3 Qualitätssicherung	4
3 Randbedingungen	4
3.1 Entwicklungsumgebung	4
3.2 Werkzeuge	4
3.3 Sprachen	4
4 Requirements and Use Cases	4
4.1 Systemebene	4
4.1.1 Stakeholder	4
4.1.2 Anforderungen	4
4.1.3 Systemkontext	4
4.1.4 Use Cases	4
4.2 Systemanalyse	4
4.3 Softwareebene	5
4.3.1 Systemkontext	5
4.3.2 Anforderungen	5
5 Design	5
5.1 System Architektur	5
5.2 Datenmodellierung	5
5.3 Verhaltensmodellierung	5
6 Implementierung	5
7 Testen	6
7.1 Testplan	6
7.2 Abnahmetest	6
7.3 Testprotokolle und Auswertungen	6
8 Lessons Learned	6
9 Anhang	6
9.1 Glossar	6
9.2 Abkürzungen	6

1 Teamorganisation

Grundsätzlich kann jedes Teammitglied eine Aufgabe seiner Wahl übernehmen. Die Aufgaben, dessen Verteilung bei jedem Meeting neu beschlossen wird, richten sich nach den zu bewältigenden Milestones(siehe [?]) zum jeweiligen Praktikumstermin. Allerdings wurden für die Projektleitung und die Pflege des RDD-Dokuments jeweils eine Person bestimmt, welche im Unterkapitel 1.1 eingesehen werden können.

1.1 Verantwortlichkeiten

Aufgabe	Zuständige/r	Bemerkung
Projektleitung	Mona	Die Projektleitung hält den Fortschritt im Auge und benachrichtigt insbesondere bei Nichteinhalten des Zeitplans alle Teammitglieder. Außerdem hat die Projektleitung bei Unstimmigkeiten immer das letzte Wort.
RDD-Pflege	Mehmet	Der Zuständige ist für die Gestaltung und für die Vollständigkeit des RDDs verantwortlich. Er kann andere Gruppenteilnehmer dazu auffordern Inhalte für das Dokument zu erarbeiten und ihm bereit zu stellen.
Protokollführung	Alle Teammitglieder	Die Protokollführung wird reihum von einem anderen Gruppenmitglied übernommen. Dabei wird folgende Reihenfolge eingehalten: <i>Mona → Marvin → Marco → Wilhelm → Mehmet → Anushavan</i>

Tabelle 1: Zuteilung von Verantwortlichkeiten

1.2 Absprachen

Listen sie hier die Absprachen im Team auf, z. B. Jour Fixe, Kommunikation, Respond-Latenz,

1.3 Repository-Konzept

Die Quelldateien sowie das RDD

2 Projektmanagement

In diesem Kapitel sollten organisatorische Punkte beschrieben und festgelegt werden.

2.1 Prozess

Legen Sie den Prozess fest, nach dem Sie das Projekt umsetzen wollen. Geben Sie ggf. grobe Schritte an, wie Planungsrunden, Sprints, oder ähnliches.

2.2 PSP/Zeitplan/Tracking

Projektstrukturplan, Ressourcenplan, Zeitplan, Abhängigkeiten von Arbeitspaketen, eventueller Zeitverzug, Visualisierung des Projektstandes, etc.

2.3 Qualitätssicherung

Überlegen sie, wie sie Qualität in ihrem Projekt sicher stellen wollen. Listen sie die Maßnahmen hier auf. Beachten sie, dass diese Maßnahmen für die unterschiedlichen Artefakte und Ebenen entsprechend unterschiedlich sein können.

3 Randbedingungen

3.1 Entwicklungsumgebung

Auflistung der Entwicklungsumgebung (Simulator, Hardware, Betriebssystem etc.)

3.2 Werkzeuge

Auflistung der im Projekt verwendeten Werkzeuge inkl. ihrer Versionen.

3.3 Sprachen

Auflistung der Programmiersprachen und Bibliotheken.

4 Requirements and Use Cases

4.1 Systemebene

Die Anforderungen aus der Aufgabenstellung sind nicht vollständig. Die Struktur der nachfolgenden Kapitel soll sie bei der Strukturierung der Analyse unterstützen. Dokumentieren Sie die Ergebnisse der Analysen entsprechend.

4.1.1 Stakeholder

Ermitteln sie die Stakeholder für das Projekt und listen sie diese hier auf.

4.1.2 Anforderungen

In der Aufgabenstellung sind Anforderungen an das System gestellt. Arbeiten sie diese hier auf und ergänzen sie diese entsprechend der Absprachen mit dem Betreuer. Achten sie auf die entsprechende Attribuierung. Berücksichtigen sie auch mögliche Fehlbedienungen und Fehlverhalten des Systems.

4.1.3 Systemkontext

Use Cases werden aus einer bestimmten Sicht erstellt. Dokumentieren sie diese mittels Kontextdiagramm oder Use Case Diagramm. Die Use Cases und Test Cases müssen zu der hier verwendeten Nomenklatur konsistent sein.

4.1.4 Use Cases

Dokumentieren sie hier, welche Use Cases sie auf der Systemebene implementieren müssen. Die Test Cases sollen später zu den Use Cases konsistent sein.

4.2 Systemanalyse

Ihr technisches System hat aus Sicht der Software bestimmte Eigenschaften. Was muss man für die Entwicklung der Software in Struktur, Schnittstellen, Verhalten und an Besonderheiten wissen? Wählen sie eine Kapitelstruktur, die am besten zur Dokumentation ihrer Ergebnisse geeignet ist.

4.3 Softwareebene

Sie sollen Software für die Steuerung des technischen Systems erstellen. Aus den Anforderungen auf der Systemebene und der Systemanalyse ergeben sich Anforderungen für Ihre Software. Insbesondere wird sich die Software der beiden Anlagenteile in einigen Punkten unterscheiden. Dokumentieren sie hier die Anforderungen, die sich speziell für die Software ergeben haben.

4.3.1 Systemkontext

Wie sieht der Kontext Ihrer Software aus? Wie erfolgt die Kommunikation mit Nachbarsystemen? Liste der ein- und ausgehenden Signale/Nachrichten.

4.3.2 Anforderungen

Welche wesentlichen Anforderungen ergeben sich aus den Systemanforderungen für ihre Software? Achten sie auf die entsprechende Attribuierung. Berücksichtigen sie auch mögliche Fehlbedienungen und Fehlverhalten des Systems.

5 Design

Anmerkung: Die Implementierung MUSS zu Ihrem Design-Modell konsistent sein. Strukturen, Verhalten und Bezeichner im Code müssen mit dem Modell übereinstimmen. Daher ist ein wohlüberlegtes Design wichtig.

5.1 System Architektur

Erstellung sie eine Architektur für Ihre Software. Geben sie eine kurze Beschreibung Ihrer Architektur mit den dazugehörigen Komponenten und Schnittstellen an. Dokumentieren sie hier wichtige technische Entscheidungen. Welche Pattern werden gegebenenfalls verwendet? Wie erfolgt die interne Kommunikation?

5.2 Datenmodellierung

Bestimmen sie das Datenmodell und dokumentieren sie es hier mit Hilfe von UML Klassendiagrammen unter Beachtung der Designprinzipien. Die Modelle können mit Hilfe eines UML-Tools erstellt werden. Hier ist dann ein Übersichtsbild einzufügen. Geben sie eine kurze textuelle Beschreibung des Datenmodells und deren wichtigsten Klassen und Methoden an.

5.3 Verhaltensmodellierung

Ihre Software muss zur Bearbeitung der Aufgaben ein Verhalten aufweisen. Überlegen sie sich dieses Verhalten auf Basis der Anforderungen und modellieren sie das Verhalten unter Verwendung von Verhaltensdiagrammen. Sie können für die Spezifikation der Prozess-Lenkung entweder Petri-Netze oder hierarchische Automaten verwenden. Die Modelle können mit Hilfe eines UML-Tools erstellt werden. Hier sind dann kommentierte Übersichtsbilder einzufügen.

6 Implementierung

Anmerkung: Nur wichtige Implementierungsdetails sollen hier erklärt werden. Code-Beispiele (snippets) können hier aufgelistet werden, um der Erklärung zu dienen. Anmerkung: Bitte KEINE ganze Programme hierhin kopieren!

7 Testen

Machen sie sich auf Basis ihrer Überlegungen zur Qualitätssicherung Gedanken darüber, wie sie die Erfüllung der Anforderungen möglichst automatisiert im Rahmen von Unit-Test, Komponententest, Integrationstest, Systemtest, Regressionstest und Abnahmetest überprüfen werden.

7.1 Testplan

Definieren sie Zeitpunkte für die jeweiligen Teststufen in ihrer Projektplanung. Dazu können sie die Meilensteine zu Hilfe nehmen.

7.2 Abnahmetest

Leiten sie die Abnahmebedingungen aus den Kunden-Anforderungen her. Dokumentieren sie hier, welche Schritte für die Abnahme erforderlich sind und welches Ergebnis jeweils erwartet wird (Test Cases).

7.3 Testprotokolle und Auswertungen

Hier fügen sie die Test Protokolle bei, auch wenn Fehler bereits beseitigt worden sind, ist es schön zu wissen, welche Fehler einst aufgetaucht sind. Eventuelle Anmerkung zur Fehlerbehandlung kann für weitere Entwicklungen hilfreich sein. Das letzte Testprotokoll ist das Abnahmeprotokoll, das bei der abschließenden Vorführung erstellt wird. Es enthält eine Auflistung der erfolgreich vorgeführten Funktionen des Systems sowie eine Mängelliste mit Erklärungen der Ursachen der Fehlfunktionen und Vorschlägen zur Abhilfe

8 Lessons Learned

Führen sie ein Teammeeting durch in dem gesammelt wird, was gut gelaufen war, was schlecht gelaufen war und was man im nächsten Projekt (z.B. im PO) besser machen will. Listen sie für die Aspekte jeweils mindestens drei Punkte auf. Weitere Erfahrungen und Erkenntnisse können hier ebenso kommentiert werden, auch Anregungen für die Weiterentwicklung des Praktikums.

9 Anhang

9.1 Glossar

Eindeutige Begriffserklärungen

9.2 Abkürzungen

Listen sie alle Abkürzungen auf, die sie in diesem Dokument benutzt haben.