

Dieses Pflichtenheft hat nur
exemplarischen Charakter.
Es sind Gliederung und Inhalt
an das eigene Projekt anzupassen.

Pflichtenheft

„Entwicklung einer Hello-world-Anwendung in Java“

Stand: 30.11.2019

Auftraggeber: Hochschule für Technik und Wirtschaft Dresden
Friedrich-List-Platz 1
D-01069 Dresden
Ansprechpartner: Prof. Dr.-Ing. habil. Dirk Müller

Auftragnehmer: Software GmbH
Musterstraße 1
D-01069 Dresden
Ansprechpartner: Dipl.-Ing. Jan Roeper

Projektname,
Version des Pflichtenheftes,
Logo des Projektes,
Auftraggeber und Auftragnehmer
mit Kontaktdaten und Firmenlogo,
Autoren etc.

Inhaltsverzeichnis

1 Zielbestimmung.....	3
1.1 Muss-Kriterien.....	3
1.2 Kann-Kriterien.....	3
1.3 Abgrenzungskriterien.....	3
2 Produkteinsatz.....	4
2.1 Anwendungsbereich.....	4
2.2 Zielgruppen.....	4
2.3 Produktumgebung.....	4
2.3.1 Architektur.....	4
2.3.2 Technologie.....	4
2.3.3 Komponenten.....	4
2.3.4 Schnittstellen.....	4
2.4 Betriebsbedingungen.....	4
3 Produktfunktionen/Anforderungen.....	5
3.1 Funktionale Anforderungen.....	5
3.1.1 Beschreibung der FA mit Rollen innerhalb der Geschäftsprozesse.....	5
3.1.2 Aktivitäten mit Benutzerschnittstelle (UI).....	6
3.1.3 Fachliches Klassendiagramm (domain model) / Produktdaten.....	7
3.2 Nichtfunktionale Anforderungen.....	7
3.2.1 Benutzbarkeit.....	7
3.2.2 Zuverlässigkeit.....	7
3.2.3 Effizienz.....	7
3.2.4 Softwarewartung.....	7
3.2.5 Sicherheit.....	7
3.2.6 Normen.....	7
4 Testung.....	7
5 Monitoring/ Support bei Übergabe oder ähnliche Leistungen.....	7
6 Dokumentation.....	8
6.1 Anwenderdokumentation.....	8
6.2 Administratordokumentation.....	8
6.3 Entwicklerdokumentation.....	8
6.4 Weitere referenzierte Dokumente.....	8
7 Vorgehen.....	8
8 Entwicklungsumgebung.....	10
9 Glossar.....	10

1 Zielbestimmung

Für das Fach Software Engineering soll eine Hello-world-Anwendung in Java entwickelt werden, um den Softwareentwicklungsprozess anhand dieser Aufgabe exemplarisch nachzuvollziehen.

Es soll ein Programm entstehen, welches nach dem Start auf dem Standardausgabestrom die Zeichenkette „hello world“ ausgibt und zum Beenden an das System den Exit-Code 0 (Null) zurückgibt.

Einleitend eine allgemeine und kurze Beschreibung des Systems und seiner Hauptaufgaben. Angabe der Gründe für Systementwicklung (z.B. Cafeteria möchte eine Bezahlapp oder ein Portal braucht eine Kommentarfunktion).

Die unbedingt zu erfüllenden Anforderungen, tabellarisch.

1.1 Muss-Kriterien

MK-IO-01	STD-Output	Implementierung von AF-IO-01: Das System muss auf dem Standardausgabestrom die Zeichenkette „hello world“ ausgeben.
MK-IO-02	STD-Input	Das System muss auf dem Standardeingabestrom keine Eingabe erwarten.
MK-BS-01	Exit-Code	Implementierung von AF-IO-01: das System muss an das Betriebssystem den Exit-Code 0 (Null) zurückgeben.
MK-SYS-01	OO-Analyse	Die Analyse des Systems muss objektorientiert erfolgen.
MK-SYS-02	UML2	Für Modellierung und Dokumentation muss UML2 genutzt werden.
MK-IMPL-01	Java Code	Die Implementierung muss in Java erfolgen.
MK-IMPL-02	Code Style	Der Java-Code ist nach den Vorgaben des Google Java Style Guide zu implementieren. (https://google.github.io/styleguide/javaguide.html)

Anforderungen, die erfüllt werden können, aber nicht zwingend erfüllt werden müssen, tabellarisch.

1.2 Kann-Kriterien

KK-BS-01	Anzeige Hilfe	Implementierung von AF-BS-01: Das System kann auf dem Standardausgabestrom bei Start mit dem Kommandozeilenparameter -h einen Hilfetext ausgeben.
----------	---------------	---

Anforderungen, die bewusst nicht erreicht werden sollen, tabellarisch.

1.3 Abgrenzungskriterien

AK-IO-01	Non-GUI	Das System soll keine grafische Bedienoberfläche haben.
AK-T-01	Testung	Das System soll keinen Usability-Test durchlaufen.

2 Produkteinsatz

In welchen Prozessen des Auftraggebers wird das System verwendet?

2.1 Anwendungsbereich

Die Hello-world-Anwendung dient der Lehre im Fach Software Engineering an der HTW Dresden. Studierende und Mitarbeiter können anhand der Anwendung den Prozess der Softwareentwicklung greifbar machen. Die Anwendung an sich hat im Ausbildungsprozess nur eine dokumentarische Funktion.

2.2 Zielgruppen

Benutzt wird die Anwendung von Studierenden und Mitarbeitern der HTW Dresden. Dabei haben diese die Rolle des Hauptnutzers. Weitere Rollen - etwa eine Administratorenrolle - treten nicht auf.

Wer verwendet das System, welche Rollen wurden analysiert ?

2.3 Produktumgebung

Das System benötigt mindestens eine installierte Java Runtime ab Java-Version 1.0. Um Java einfach starten zu können, sollte die Pfad-Variable auf den bin-Ordner der Javaumgebung gesetzt sein. Hardwareanforderungen bestehen keine.

2.3.1 Architektur

2.3.2 Technologie

2.3.3 Komponenten

2.3.4 Schnittstellen

Beschreibung von Soft-, Hard- und Orgware mit Versionsangaben und technischen Parametern ohne Implementierungsdetails. Je nach Komplexität gesonderte Beschreibung der Architektur, spezieller Technologien, Komponenten und externer Schnittstellen. (Text, Tabellen und UML2-Diagramme)

Je nach System müssen Unterpunkte ergänzt oder können weggelassen werden.

2.4 Betriebsbedingungen

Das System wird für die Anwendungsfälle AF-IO-01 und AF-BS-01 in den Computerlaboren der HTW Dresden verwendet. Hauptbenutzer bedienen das Kommandozeilenprogramm an den dort aufgestellten Rechnern unter dem jeweiligen persönlichen Login im Konsolenfenster.

Beschreibung von Bedienung, Umfeld, Einsatzart, Betriebszeiten etc.

Ein unbeaufsichtigter Betrieb ist nicht vorgesehen. Die Rechner der Labore werden zyklisch mit aktueller Hardware erneuert, als Betriebssysteme kommen Linux und Windows zum Einsatz. Die Installation von Betriebssystem und Software erfolgt in der Semesterpause durch den Laborbereich. Die Räume sind klimatisiert. Die Rechner haben keine unterbrechungsfreie Stromversorgung (USV).

3 Produktfunktionen/Anforderungen

Die funktionalen Anforderungen sind dem Lastenheft zu entnehmen und bei Bedarf anzupassen oder zu konkretisieren. (Tabellen und UML2-Diagramme)

3.1 Funktionale Anforderungen

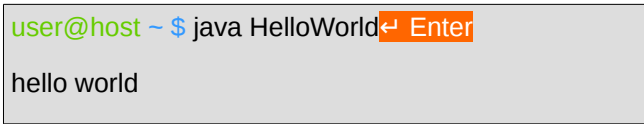
3.1.1 Beschreibung der FA mit Rollen innerhalb der Geschäftsprozesse

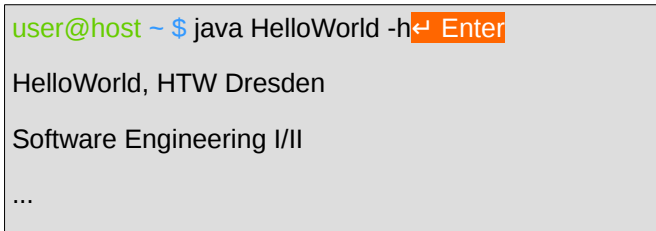
Im exemplarischen Prozess der Ausbildung für das Fach Software Engineering hat das Programm nur zwei Anwendungsfälle: die Hauptfunktion und eine Hilfsfunktion. Der angemeldete Nutzer führt das Programm in der Konsole aus.

AF-01	Anzeige des Hello-world-Textes	Ein am Betriebssystem angemeldeter Benutzer führt in einer geöffneten Konsole das Programm ohne Kommandozeilenparameter aus. Der Aufruf erfolgt mittels Java-Interpreter. Der Text „hello world“ wird auf dem Standardausgabestrom ausgegeben. Das Programm gibt den Exit-Code 0 zurück.
AF-02	Anzeige des Hilfe-Textes	Ein am Betriebssystem angemeldeter Benutzer führt in einer geöffneten Konsole das Programm mit Kommandozeilenparameter „-h“ aus. Der Aufruf erfolgt mittels Java-Interpreter. Ein Hilfetext wird auf dem Standardausgabestrom ausgegeben. Das Programm gibt den Exit-Code 0 zurück.

3.1.2 Aktivitäten mit Benutzerschnittstelle (UI)

Für die obigen Anwendungsfälle werden die Abläufe beschrieben mit den dazugehörigen Aktivitäten und prototypischer Darstellung der Benutzerschnittstelle UI/GUI. Abläufe mit Nutzer, Zielen, Daten (Objektflüssen), Bedingungen, Ereignissen, Alternativen beschreiben. (Tabellen, UML2-Diagramme und Grafiken)

Anwendungsfall ID	AF-01
AF Name	Anzeige des Hello-world-Textes
Akteur	am Betriebssystem angemeldeter Nutzer
Vorbedingung	Systemkonsole geöffnet und Pfad-Variable auf den bin-Ordner der Javaumgebung gesetzt
Auslösendes Ereignis	Bestätigung der Konsoleneingabe <code>java HelloWorld</code> mit ENTER
Nachbedingung Erfolg	Programmausgabe „hello world“ erfolgt auf Standardausgabestrom.
Nachbedingung Fehlschlag	Programm konnte nicht gestartet werden
Ablauf	<ul style="list-style-type: none"> - <code>java HelloWorld</code> in Konsole eingeben - ENTER-Taste - Textausgabe erfolgt
Benutzerschnittstelle	

Anwendungsfall ID	AF-02
AF Name	Anzeige des Hilfe-Textes
Akteur	am Betriebssystem angemeldeter Nutzer
Vorbedingung	Systemkonsole geöffnet und Pfad-Variable auf den bin-Ordner der Javaumgebung gesetzt
Auslösendes Ereignis	Bestätigung der Konsoleneingabe <code>java HelloWorld -h</code> mit ENTER
Nachbedingung Erfolg	Ausgabe des Hilfetextes erfolgt auf Standardausgabestrom.
Nachbedingung Fehlschlag	Programm konnte nicht gestartet werden
Ablauf	<ul style="list-style-type: none"> - <code>java HelloWorld</code> in Konsole eingeben - ENTER-Taste - Textausgabe erfolgt
Benutzerschnittstelle	

3.1.3 Fachliches Klassendiagramm (domain model) / Produktdaten

Für die HelloWorld-Anwendung sind keine Daten dauerhaft zu speichern.

Beschreibung der Domäne aus fachlicher Sicht anhand eines Klassendiagramms, dabei sollen keine Implementierungsdetails festgelegt werden.

Die Klassen können u.a. den Anwendungsfällen und deren Aktivitäten entnommen werden.

Angabe, welche Klassen zu persistieren sind. Prüfen, ob Klassenbezeichner im Glossar aufzuführen sind.

(Der Beispieltext fällt nur hier für dieses Hello-World-System so knapp aus.)

3.2 Nichtfunktionale Anforderungen

3.2.1 Benutzbarkeit

3.2.2 Zuverlässigkeit

3.2.3 Effizienz

3.2.4 Softwarewartung

3.2.5 Sicherheit

3.2.6 Normen

Beschreibung der nichtfunktionalen Anforderungen in den aufgeführten Unterpunkten.

Je nach System müssen Unterpunkte ergänzt oder können weggelassen werden.

NF-B1	Benutzung	HelloWorld soll nur per Konsole genutzt werden.
NF-E1	Effizienz	Die Ausgabe der Texte soll unmittelbar erfolgen.
NF-W1	Softwarewartung	Es ist langfristig vorgesehen, die Texte in verschiedenen Übersetzungen anzuzeigen.
NF-S1	Sicherheit	Diesbezüglich liegen keine Anforderungen vor.
NF-N1	Normen	Die Anwendung braucht keine besonderen Normen zu erfüllen.

4 Testung

Es wird ein einfacher Funktionstest für den Anwendungsfall AF-01 durchgeführt.

Beschreibung der vereinbarten Teststufen und Testarten und die vorgesehene Durchführung.

5 Monitoring/ Support bei Übergabe oder ähnliche Leistungen

Im Rahmen des semesterbegleitenden Supports für die HelloWorld-Anwendung werden folgende Leistungen erbracht:

- Bereitstellung des Repositories,
- individueller Support in den Praktika,
- Rufbereitschaft 8x5 per E-Mail

Je nach Vertrag sollen in diesem Kapitel zusätzlich zu erbringende Leistungen beschrieben werden.

6 Dokumentation

Beschreibung der im Rahmen des Projektes zu erstellenden Dokumente, die für die Benutzung, Administration und Weiterentwicklung der Software an den AG zu übergeben sind.

6.1 Anwenderdokumentation

Die Anwenderdokumentation wird als typische liesmich.txt Datei in deutscher und als readme.txt englischer Sprache im Repository zur Verfügung gestellt.

6.2 Administratordokumentation

Eine Administratordokumentation ist nicht vorgesehen.

6.3 Entwicklerdokumentation

Als Entwicklerdokumentation werden die mit javadoc generierten HTML-Dokumente im Repository zur Verfügung gestellt.

6.4 Weitere referenzierte Dokumente

Verweise auf Repository, Lastenheft, Projektwiki, Projektplan, Präsentation, etc.

Das Pflichtenheft wurde mit Bezug auf das „Lastenheft - Entwicklung einer Hello-world-Anwendung in Java“ erstellt. Lastenheft, Pflichtenheft und die Anwender-, Administrator- und Entwicklerdokumentation befinden sich im Repository Software Engineering.

7 Vorgehen

Wie?

Für den Anwendungsfall AF-01 und AF-02 wird ein Prototyp erstellt, der gemäß der nicht funktionalen Anforderungen inkrementell erweitert wird. Danach erfolgt der Funktionstest. Diese letzte Testversion gilt als Release Candidate auf deren Basis auch die Dokumentation abgeschlossen wird (code freeze). Anschließend erfolgt die Übergabe.

Entwicklungsplan, Vorgehensmodell, Meilensteine und Fortschrittskontrolle, Release-Planung.

(Meilensteine sind die für die Praktika in SE I und SE II als Aufgaben gestellten Themen und Termine.)

Meilensteine sind:

Datum	Meilenstein
27.09.2019	Auftakt
28.09.2019	Projektplan und Pflichtenheft
01.10.2019	Erstellen der Entwicklungsumgebung
03.10.2019	Prototyp
05.10.2019	Funktionstest
07.10.2019	Release Candidate
08.10.2019	Übergabe

Die Fortschrittskontrolle erfolgt anhand folgender Indikatoren:

Indikator		Kick-Off	Projektplan und Pflichtenheft	Erstellen der Entwicklungs- umgebung	Prototyp	Funktions- test	Release Candidate	Übergabe
Pflichtenheft [% erledigte Gliederungs- Punkte]	Soll							
	Ist							
Umgebung [Tools]	Soll							
	Ist							
Diagramme [Anzahl]	Soll							
	Ist							
Quellcode [LOC]	Soll							
	Ist							
Verhältnis [LOC/ Kommentare]	Soll							
	Ist							
Tests [Fälle/ Methoden]	Soll							
	Ist							
Anwenderdoku [Wörter]	Soll							
	Ist							
Entwicklerdoku [Wörter]	Soll							
	Ist							
Release [Artefakte]	Soll							
	Ist							

Für die Fortschrittskontrolle existieren verschiedene Verfahren. Parameter, Granularität und Ebenen zur Beurteilung des Projektstandes auf Basis von Soll/Ist-Vergleichen müssen geeignet gewählt werden. Diagramme, Tabellen etc. müssen regelmäßig aktualisiert und fortgeschrieben werden.

8 Entwicklungsumgebung

Womit?

Beschreibung der Entwicklungsumgebung, gegliedert nach Hardware, Software und Orgware.

Für die Entwicklung dieses Systems werden ein Texteditor und ein Java-Compiler benötigt. Verwendet wurde GNU nano 2.2.6 und javac 1.8.0_91.

Aufgrund der Einfachheit des Produktes erfolgte nur ein Funktionstest ohne Testwerkzeug.

Die Entwicklerdokumentation wurde mit javadoc erstellt, der Quellcode ist entsprechend kommentiert.

An die Hardware und Orgware bestehen keine besonderen Anforderungen.

9 Glossar

AG: Auftraggeber

AN: Auftragnehmer

FA: Funktionale Anforderungen

UI: User Interface (Benutzerschnittstelle), hier Ein- und Ausgabe von Text in der Konsole

Erklärung unbekannter oder unscharfer Begriffe der Domäne des AG und der technischen Domäne des AN. Begriffe sind unbedingt konsistent zu verwenden.

Bei umfangreichen Projekten kann ein spezielles Wörterbuch erstellt werden.