



<TITEL DES PRAKTIKUMS>

<TITEL DES TEILPROJEKTES>

Software-Entwicklungspraktikum (SEP)
Sommersemester 2017

Pflichtenheft

Auftraggeber
Technische Universität Braunschweig
<Name des Instituts>
<Name des Institutsleiters>
<Straße und Hausnummer>
<Postleitzahl und Ort>

Betreuer: <Name>

Auftragnehmer:

Name	E-Mail-Adresse
<Name des Teilnehmers>	<E-Mail-Adresse>
<Name des Teilnehmers>	<E-Mail-Adresse>
<Name des Teilnehmers>	<E-Mail-Adresse>
...	...

Braunschweig, 29. März 2017

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1
1.1
1.2
2
3
3.1
3.2
4
4.1
4.2
4.3
5
5.1
5.2
5.3
6
6.1
6.2
6.3
7

Inhaltsverzeichnis

1 Zielbestimmung	5
1.1 Musskriterien	5
1.2 Sollkriterien	5
1.3 Kannkriterien	6
1.4 Abgrenzungskriterien	6
2 Produkteinsatz	7
2.1 Anwendungsbereiche	7
2.2 Zielgruppen	7
2.3 Betriebsbedingungen	7
3 Produktübersicht	9
4 Produktfunktionen	11
5 Produktdaten	13
6 Nichtfunktionale Anforderungen	14
6.1 Funktionalität	14
6.2 Sicherheit	14
6.3 Benutzbarkeit	15
6.4 Änderbarkeit	15
6.5 Qualitätsanforderungen	15
7 Benutzeroberfläche/Schnittstellen	17
8 Technische Produktumgebung	19
8.1 Software	19
8.2 Hardware	19
8.3 Produktschnittstellen	19
9 Glossar	21

Abbildungsverzeichnis

3.1	Use-Case-Diagramm Buchungssystem	9
7.1	Login $\langle UI10 \rangle$	18

1 Zielbestimmung

Dieser Abschnitt hat die Aufgabe, als eine Art Einleitung zu dienen. Es soll ein kurzer Umriss über Ziel und Motivation des Gesamt- und ggf. der Teilprojekte dargestellt werden. Beschrieben wird die Hauptaufgabe des Systems. Wichtig ist, den Grund für die Systementwicklung (Probleme oder Geschäftsidee) und damit ihre Ziele herauszuarbeiten. (Umfang ca. $\frac{1}{2}$ – 2 Seiten)

Hinweis zu den Templates:

Dieses Template enthält Hinweise und Beispiele, die selbstverständlich zu entfernen sind. Angaben in <...> sind mit dem entsprechendem Text zu füllen.

1.1 Musskriterien

Musskriterien: unabdingbare Leistungen der Software.

Hier wird aufgeführt, welche Funktionalitäten/Leistungen das Softwareprodukt in jedem Fall erfüllen muss, damit es genutzt werden kann.

<RM1> Das ist wichtig.

<RM2> Und das ist auch wichtig.

1.2 Sollkriterien

Sollkriterien: erstrebenswerte Leistungen

Dies sind Kriterien, die für die Lauffähigkeit des Produkts nicht zwingend erforderlich sind, für die Erreichung der Projektziele aber erfüllt werden sollten.

<RS1> Das sollte auch realisiert werden.

<RS2> Und das ebenfalls.

1.3 Kannkriterien

Kannkriterien: Leistungen die enthalten sein können, denen der Auftraggeber jedoch neutral gegenüber steht. Die Erfüllung dieser Kriterien ist nicht unbedingt notwendig, sie sollten nur angestrebt werden, falls noch ausreichend Kapazitäten vorhanden sind.

⟨RC1⟩ Wenn noch Zeit ist, wäre diese Funktion wünschenswert.

⟨RC2⟩ Das würde ich mir wünschen, ist aber auch nicht so wichtig.

1.4 Abgrenzungskriterien

Abgrenzungskriterien: Leistungen die explizit nicht umgesetzt werden.

Hier ist zu verdeutlichen, welche Ziele mit dem Produkt bewusst nicht erreicht werden sollen oder können. Speziell sind hier Funktionen zu erwähnen, die sich der Kunde ursprünglich gewünscht (oder genannt) hat, die aber, nach Einigung, doch nicht umgesetzt werden sollen. Auch Funktionen, die im Allgemeinen von ähnlichen Systemen zu erwarten wären, hier aber explizit nicht umzusetzen sind (z.B. Login in einem Forum), sollten erwähnt werden. Zu jedem System gehört normalerweise auch ein Benutzerhandbuch. Wird ein Handbuch nicht benötigt, sollte dies hier festgehalten werden, sonst kann der Kunde später ein Handbuch verlangen.

⟨RW1⟩ Das will ich auf gar keinen Fall haben.

⟨RW2⟩ Dafür bezahle ich nicht.

⟨RW3⟩ Das wird schon durch ein anderes System abgedeckt.

2 Produkteinsatz

Dieser Abschnitt hat die Aufgabe, den Einsatzbereich, die Zielgruppen und die Betriebsbedingungen des zu entwickelnden Systems klarzustellen. (Umfang mit Unterkapiteln ca. 1,5 - 4 Seiten)

2.1 Anwendungsbereiche

Hier wird der Bereich beschrieben, in dem das Produkt eingesetzt werden soll, bzw. Bereiche, für die das Produkt nicht gedacht ist. Zum Beispiel könnte hier beschrieben werden, dass ein zu entwickelndes Bibliothekssystem nur für den Einsatz in einer speziellen Bibliothek des Kunden und nicht für öffentliche Bibliotheken geeignet ist. Natürlich muss dann auch die Bibliothek des Kunden (und deren Besonderheiten) beschrieben werden.

2.2 Zielgruppen

Hier wird angegeben, für welche Anwender (z. B. Sekretärin, andere Entwickler) das Produkt im Wesentlichen gedacht bzw. nicht gedacht ist. Zum Beispiel:

- Sekretärin: Beschafft Bücher und legt diese in der Bibliothek an.
- wiss. Mitarbeiter: Leiht Bücher aus und gibt diese zurück.
- Student: ...

2.3 Betriebsbedingungen

Hier werden die unterschiedlichen Bedürfnisse und Anforderungen an das Produkt aufgelistet. Dies können folgenden Punkte sein:

- physikalische Umgebung des Systems (z. B. Büroumgebung, mobiler Einsatz)
- tägliche Betriebszeit (z. B. Dauerbetrieb)

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

- ständige Beobachtung des Systems durch einen Bediener oder unbeaufsichtigter Betrieb

3 Produktübersicht

Dieser Abschnitt hat die Aufgabe, die Funktionalität des zu entwickelnden Systems grafisch mit Hilfe von Use-Case-Diagrammen und einer kurzen verbalen Beschreibung zu charakterisieren. Im Erklärungstext sollte darauf eingegangen werden, welcher Use-Case welches Kriterium erfüllt. Es sind so viele Use-Case-Diagramme einzufügen, wie zur vollständigen und übersichtlichen Beschreibung der Systemfunktionalität notwendig sind.

Anmerkungen zu Use-Cases:

- Der Zweck eines Use-Cases ist es, zu beschreiben wie ein Akteur das System benutzen kann, um ein bestimmtes Ziel zu erreichen.
- Ein Use-Case ist nicht immer eine Funktion, sondern das Ziel des Systems und kann mehrere Funktionen umfassen. In diesem Fall, kann man dann die einzelnen Funktionen in Aktivitätsdiagrammen beschreiben.
- Use-Cases werden klassischerweise so benannt, wie die Ziele aus Sicht der Akteure heißen.
- Das Diagramm soll nur den groben Aufbau des Systems beschreiben, damit man sehen kann, ob das System das Richtige tut oder nicht.

Beispiel:

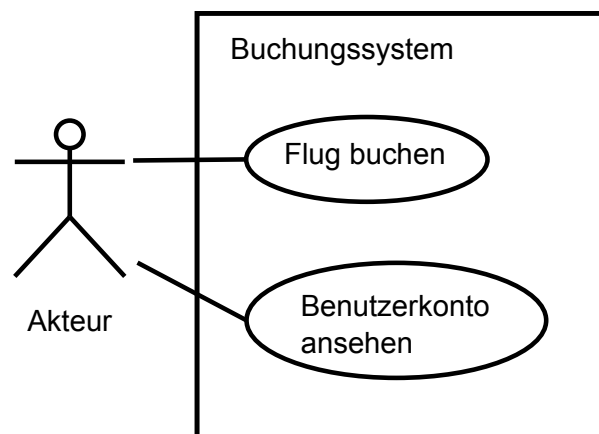


Abbildung 3.1: Use-Case-Diagramm Buchungssystem

Dieses Diagramm zeigt ein minimales Use-Case-Diagramm für ein Buchungssystem. Der Akteur, hier ein Benutzer der einen Flug buchen möchte, kann einen Flug buchen oder sein Benutzerkonto ansehen. Diese beiden Use-Cases sind für den Benutzer zielführend. Ein Use-Case *Flug suchen* wäre hier falsch, denn das ist nicht das Ziel eines Buchungssystems. Der Use-Case *Flug buchen* beinhaltet mehrere Funktionen des zugehörigen Systems. Diese wären *Einloggen*, *Flug reservieren* usw. und sollen im nächsten Kapitel beschrieben werden. In diesem Fall wäre nun ein Aktivitätsdiagramm zur Beschreibung der einzelnen Funktionen notwendig.

Interessante Use-Cases können mit einem Aktivitätsdiagramm genauer erläutert werden. **Es soll mindestens ein Aktivitätsdiagramm für einen nicht trivialen Use-Case erstellt werden.**

4 Produktfunktionen

In Abhängigkeit von den gewählten Konzepten erfolgt hier eine Konkretisierung und Detaillierung der Funktionen aus den Use-Case-Diagrammen und ggf. dem Angebot.

Wurde beispielsweise im Angebot die Funktionalität durch verbal beschriebene Geschäftsprozesse definiert, dann kann hier eine Detaillierung erfolgen. Mögliche Gliederung der Anwendungsfälle:

- Anwendungsfall
- Listen
- Reports

Die Produktfunktionen müssen die Kriterien aus den Zielbestimmungen abdecken. Dabei kann es je nach Kriterium eine oder mehrere Funktion geben. Jedes Kriterium muss von mindestens einer Funktion erfüllt werden. Die nachfolgende Gliederung bitte für alle Funktionen übernehmen:

Beispiel:

Lagerverwaltung $\langle F10 \rangle$

Anwendungsfall: Automatisches Einlagern

Anforderung: $\langle LF20 \rangle$ (Wenn kein Lastenheft vorhanden, dann Kriterium aus den Zielbestimmungen, z.B.: $\langle RM1 \rangle$)

Ziel: Ein Reifen erscheint am Systemeingang (Scanner), erhält einen Lagerplatz zugewiesen und wird dort eingelagert.

Vorbedingung: Das Scannen des Barcode-Reifens muss erfolgreich sein, sonst kann der Typ nicht ermittelt werden. Solche unbekannten Reifen werden direkt in den Überlauf gefördert.

Nachbedingung Erfolg: Reifen ist physikalisch eingelagert und logisch in der Datenbank verbucht.

Nachbedingung Fehlschlag: Der Reifen wurde infolge gestörter Fördermechanik nicht eingelagert (liegt im Überlauf) oder produzierte aufgrund inkonsistenter Datenbank einen „Platz belegt“ - Fehler beim Anfahren eines irrtümlich als frei angenommenen Platzes.

Akteure: Produktion

Auslösendes Ereignis: SPS meldet der Steuerung, dass am Eingangsscanner ein Reifen mit Seriennummer X des Typs Y eingetroffen ist.

Beschreibung:

1. Reifentypinformationen ermitteln (besonders Höhe des Reifens bei Wahl zwischen unterschiedlich hohen Lagerplätzen wichtig).
2. Alle Module ermitteln, die
 - Platz auf den Einlagerstichen haben
 - momentan nicht im Störungszustand sind
 - freie Lagerplätze in der geforderten Höhe aufweisen.
3. Lagerplatz nach Gleichverteilungsgrundsatz bestimmen.
4. Reifen auf den Einlagerstich des gewählten Moduls befördern.
5. Sobald er auf dem vordersten Platz des Einlagerstichs steht, dem Modul den Befehl zur Reifenaufnahme und Einlagerung auf den gewählten Platz schicken.

Erweiterung: (optional)

2a Zur Effizienzsteigerung auch Module ansteuern, die momentan keinen Platz auf den Einlagerstichen haben, aber wahrscheinlich so schnell einlagern, dass der Reifen nach der Fahrtzeit zum Modul auf den Stich eingelagert werden kann (Überwachung des „Unterwegsbestandes“ an Reifen für ein bestimmtes Modul).

3a Lagerplatz des Reifens möglichst nah zum Einlagerstich im RBG wählen (kürzere RBG-Fahrtzeiten).

Alternativen: (optional)

2a Wenn kein Lagerplatz gefunden wird, Reifen zum Überlauf schicken (der Einlagerförderer wird niemals angehalten!).

Login <F20>

Anwendungsfall: ...

5 Produktdaten

Die langfristig zu speichernden Daten sind aus Benutzersicht detaillierter zu beschreiben. Dabei bietet sich eine formale Beschreibung an, um eine größere Präzisierung zu erreichen. Es sollte eine Menge an erwarteten Daten angegeben werden.

Es kann die Darstellung gemäß Beispiel verwendet werden (alternativ kann auch ein Klassendiagramm mit entsprechender Beschreibung erstellt werden):

Lagerdaten $\langle D10 \rangle$

Daten der Lagerplätze (max. 5.000):

- Modulnummer,
- Regalseite,
- Regalspalte,
- Regalzeile,
- Fachhöhe,
- Platzsperre (0 = nicht gesperrt, 1 = gesperrt für Einlagerung, 2 = gesperrt für Auslagerung, 3 = gesperrt für alle Zugriffe),
- Reifenstatus (0 = frei, 1 = reserviert für Einlagerung, 2 = belegt, 3 = reserviert für Auslagerung),
- Reifenseriennummer.

Moduldaten $\langle D20 \rangle$

Daten der Module (max. 20):

- Modulnummer,
- Sperrkennzeichen (0 = nicht gesperrt, 1 = gesperrt für Einlagerung, 2 = gesperrt für Auslagerung, 3 = gesperrt für alle Zugriffe),
- maximale Kapazität,
- freie Kapazität,
- belegte Plätze (ergibt sich aus Status und Zahl der zugeordneten Lagerplätze, wird aus Geschwindigkeitsgründen allerdings redundant mitgeführt).

6 Nichtfunktionale Anforderungen

In diesem Kapitel wird festgelegt, welche Qualitätsmerkmale das zu entwickelnde Produkt in welcher Qualitätsstufe besitzen soll. Anschließend werden die als am wichtigsten bezeichneten Qualitätsmerkmale operationalisiert, d.h. in konkrete Produktanforderungen detailliert, falls sie nicht als allgemeine Richtlinie (z. B. Standard, Norm) zur Verfügung gestellt werden können.

Sofern projektspezifisch möglich, ist die Anordnung gemäß der vorliegenden Tabellen zu nutzen. Ggf. können aber auch andere Qualitätsanforderungen gestellt oder nicht zutreffende weggelassen werden.

6.1 Funktionalität

Produktqualität	sehr gut	gut	normal	nicht relevant
Angemessenheit		x		
Richtigkeit	x			
Interoperabilität		x		
Ordnungsmäßigkeit	x			

Hier sollte dann darauf eingegangen werden, wieso ihr die einzelnen Punkte entsprechend markiert habt.

6.2 Sicherheit

Produktqualität	sehr gut	gut	normal	nicht relevant
Zuverlässigkeit				x
Reife	x			
Fehlertoleranz			x	
Wiederherstellbarkeit	x			

Hier sollte dann darauf eingegangen werden, wieso ihr die einzelnen Punkte entsprechend markiert habt.

6.3 Benutzbarkeit

Produktqualität	sehr gut	gut	normal	nicht relevant
Verständlichkeit		x		
Erlernbarkeit			x	
Bedienbarkeit		x		
Effizienz				
Zeitverhalten		x		
Verbrauchsverhalten		x		

Hier sollte dann darauf eingegangen werden, wieso ihr die einzelnen Punkte entsprechend markiert habt.

6.4 Änderbarkeit

Produktqualität	sehr gut	gut	normal	nicht relevant
Analysierbarkeit		x		
Modifizierbarkeit		x		
Stabilität	x			
Prüfbarkeit		x		
Übertragbarkeit				
Anpassbarkeit				x
Installierbarkeit		x		
Konformität		x		
Austauschbarkeit			x	

Hier sollte dann darauf eingegangen werden, wieso ihr die einzelnen Punkte entsprechend markiert habt.

6.5 Qualitätsanforderungen

Die oben als am wichtigsten bezeichneten Qualitätsmerkmale werden im Folgenden operationalisiert, d.h. in konkrete Produkthanforderungen detailliert oder es wird angegeben, welche Richtlinie (z. B. Standard, Norm) einzuhalten ist. Diese Qualitätsanforderungen werden wie im Beispiel definiert. Zu prüfen ist, ob die gewünschte Qualität mit den in Produktdaten genannten Datenmengen erreicht werden kann.

Beispiele:

- $\langle Q10 \rangle$ Die Funktion $\langle F20 \rangle$ darf nicht länger als 5 Sekunden Antwortzeit benötigen.
- $\langle Q20 \rangle$ Alle Reaktionszeiten auf Benutzeraktionen müssen unter 2 Sekunden liegen (außer Funktion $\langle F20 \rangle$).
- $\langle Q30 \rangle$ Die im Rahmen der automatischen Einlagerung $\langle F10 \rangle$ notwendige Platzwahl für einen am Anmeldescanner gemeldeten Reifen darf aus Gründen der Kommunikation mit der SPS nicht länger als 3 Sekunden dauern, ansonsten kann die SPS die Lieferung des Reifens zum richtigen Modul nicht garantieren.
- $\langle Q40 \rangle$ Das Produkt soll plattformunabhängig sein.
- $\langle Q50 \rangle$ Das Produkt muss anwenderfreundlich sein. (Intuitive Bedienbarkeit für Benutzer ohne EDV-Vorkenntnisse, umfangreiche Hilfefunktion)
- $\langle Q60 \rangle$ Die Produkt soll fehlertolerant bezüglich Bedien- und Eingabefehler sein.

7 Benutzeroberfläche/Schnittstellen

In diesem Kapitel soll, an euer Projekt angepasst, die Benutzeroberfläche und/oder die Schnittstellen beschrieben werden.

Benutzeroberfläche:

Es sollen die grundlegenden Anforderungen an die Benutzeroberfläche festgelegt werden, z. B. Fensterlayout, Dialogstruktur und Mausbedienung. Die Festlegungen sollen sich auf die produktspezifischen Ausprägungen beschränken. Details werden durch Prototypen und Pilotsysteme spezifiziert (ggf. im Anhang des Pflichtenhefts).

Gibt es verschiedene Rollen, die das Produkt benutzen (z. B. Sachbearbeiter, Administrator), dann sind für jede Rolle die Zugriffsrechte und die davon abhängigen sichtbaren Benutzeroberflächen (Menüs, Fenster, Benutzerdialoge, ...) aufzuführen.

Wichtig: Es sollen hier bereits ein oder zwei Mockups eurer Benutzeroberfläche gezeigt werden.

Die einzelnen Anforderungen werden analog zu den Funktionsanforderungen nummeriert, allerdings mit den vorgesetzten Buchstaben UI.

Beispiele:

Standardmäßig sind das Windows-Gestaltungs-Regelwerk sowie die Norm ISO 9241-10: 1996 (Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten, Teil 10: Grundsätze der Dialoggestaltung) in allen Benutzeroberflächen zu beachten.

Folgende Rollen sind zu unterscheiden:

Rolle	Rechte	Benutzeroberfläche
Instandhaltung	$\langle F10 \rangle$, $\langle F20 \rangle$	Funktionsspezifische Eingabemasken, ...
Werksleitung	$\langle F10 \rangle$...
...

Login $\langle UI10 \rangle$

Kurze Beschreibung siehe Abbildung 7.1.

Suche $\langle UI20 \rangle$

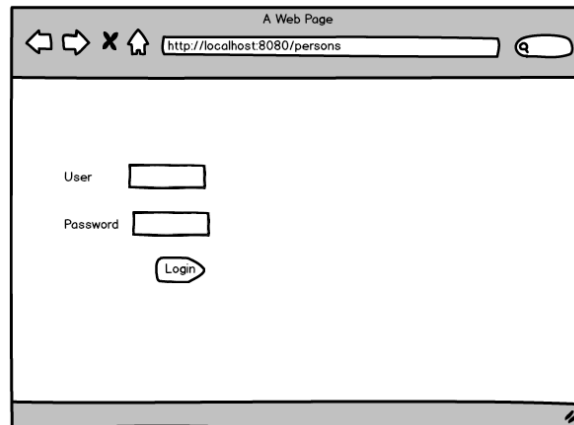


Abbildung 7.1: Login $\langle UI10 \rangle$

...

Schnittstellen:

Bei Produkten, die keine Benutzeroberfläche haben, werden hier analog die Schnittstellenkonventionen beschrieben, die für das anwendende System wichtig sind. Es sollen also die Schnittstellen spezifiziert werden, die eure Anwendung zur Verfügung stellt.

8 Technische Produktumgebung

In diesem Kapitel wird die technische Umgebung des Produktes beschrieben. Bei Client/Server-Anwendungen ist die Umgebung jeweils für Client und Server getrennt anzugeben.

8.1 Software

Hier wird angegeben, welche Softwaresysteme (z. B. Betriebssystem, Datenbank, Fenstersystem, usw.) zur Verfügung stehen.

Beispiel:

Server-Betriebssystem: Linux (Debian 7)

Client-Betriebssystem: Windows 7

8.2 Hardware

Hier werden die Hardware Komponenten (z. B. CPU, Peripherie) in minimaler und maximaler Konfiguration aufgeführt, die für den Produkteinsatz vorgesehen sind.

Beispiel:

Server: Virtueller Server mit 4 Cores und 4 GB RAM

Client: Standard PC und browserfähiges Gerät mit Grafikbildschirm (für Fernwartung)

8.3 Produktschnittstellen

Wird das Produkt in eine bestehende oder geplante Produktfamilie eingeordnet, so werden hier die entsprechenden Schnittstellen definiert.

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

Beispiel:

Die Kommunikation mit der unterlagerten SPS erfolgt über getrennt definiertes (eigenes Pflichtenheft) TCP/IP - Protokoll. Analoges gilt für die Kommunikation mit dem Testmaschinen-Rechner.

9 Glossar

Hier werden Fachbegriffe erklärt.