



<TITEL DES PRAKTIKUMS>

<TITEL DES TEILPROJEKTES>

Software-Entwicklungspraktikum (SEP)
Sommersemester 2017

Testspezifikation

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

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1
2
2.1
2.2
2.3
2.4
2.5
3
3.1
3.2
3.3
4
4.1
4.2
4.3
5
5.1
5.2
5.3
6

Inhaltsverzeichnis

1	Einleitung	4
2	Testplan	5
2.1	Zu testende Komponenten	5
2.2	Zu testende Funktionen/Merkmale	5
2.3	Nicht zu testende Funktionen	6
2.4	Vorgehen	6
2.5	Testumgebung	7
3	Abnahmetest	8
3.1	Zu testende Anforderungen	8
3.2	Testverfahren	8
3.2.1	Testskripte	8
3.3	Testfälle	9
3.3.1	Testfall <T100> - Lager anlegen	10
3.3.2	Testfall <T200> - Lager löschen	12
4	Integrationstest	13
4.1	Zu testende Komponenten	13
4.2	Testverfahren	13
4.2.1	Testskripte	13
4.3	Testfälle	13
4.3.1	Testfall <T200> - Komponente x+y	14
5	Unit-Tests	15
5.1	Zu testende Komponenten	15
5.2	Testverfahren	15
5.2.1	Testskripte	15
5.3	Testfälle	15
5.3.1	Testfall <T300> - Klasse x	16
6	Glossar	17

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

Abbildungsverzeichnis

1 Einleitung

Wie in der Vorlesung erwähnt, ist das Testen von Software unerlässlich und darf bei keinem Softwareentwicklungsprozess fehlen.

Diese Testdokumentation ist angelehnt an den IEEE 829-Standard, der als der bekannteste Standard für Software-Testdokumentationen gilt. Der Standard definiert eine Menge von Testdokumenten und beschreibt deren Inhalte.

In diesem Kapitel soll kurz allgemein beschrieben werden, welche Software getestet wird, welchen Qualitätsanforderungen sie genügt und welche Standards unter Umständen zu beachten sind. (Umfang ca. $\frac{1}{3}$ bis 1 Seite)

Hinweis zu den Templates:

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

Kapitel, die bereits in der Abnahmetestspezifikation bearbeitet wurden, müssen hier nicht erneut bearbeitet werden. Es sollten jedoch die Annotationen umgesetzt worden sein. Es sollen also Kapitel 4-5 neu erarbeitet und Kapitel 2.1 überarbeitet werden.

Dieses Kapitel kann aus der Abnahmetestspezifikation übernommen werden, sollte jedoch die Bearbeitung der Annotationen beinhalten.

2 Testplan

Der Testplan ist das zentrale Dokument der Qualitätssicherung und wird daher frühzeitig erstellt. Hier wird Umfang und Vorgehensweise der Qualitätssicherung beschrieben. Außerdem werden Testgegenstände und deren zu testenden Eigenschaften bzw. Funktionen identifiziert. Ferner werden die durchzuführenden Maßnahmen und die dafür verantwortlichen Personen definiert. Falls erforderlich, sollte hier auch auf allgemeine Risiken eingegangen werden.

Dieses Kapitel kann aus der Abnahmetestspezifikation übernommen werden, sollte jedoch die Bearbeitung der Annotationen beinhalten. Außerdem sollte das Unterkapitel 2.1 der zu testenden Komponenten überarbeitet werden.

2.1 Zu testende Komponenten

Hier sind sämtliche zu testenden Objekte einschließlich der Versionsnummer aufzuführen. Ebenso ist anzugeben, auf welchem Medium die Software vorliegt, ob dies einen Einfluss auf Hardwareanforderungen hat und ob die Software vor Testbeginn in irgendeiner Weise transformiert werden muss. Außerdem wird auf zum Objekt gehörende Dokumentation der Komponente (Lasten-, Pflichtenheft, später auch Systementwurf) referenziert.

Anmerkung:

Dieses Dokument wird am Ende noch einmal zusammen mit den Testprotokollen in der Testspezifikation abgegeben. Da vorher im Systementwurf die Komponenten benannt und nummeriert wurden, sollten diese für die zweite Abgabe erweitert werden, so dass eine Verbindung zwischen dem Systementwurf und diesem Dokument entsteht.

2.2 Zu testende Funktionen/Merkmale

Dieser Punkt beinhaltet alle Eigenschaften bzw. Funktionen und deren Kombinationen, die zu testen sind.

Sämtliche Funktionalitäten, die getestet werden sollen, werden hier aufgeführt. Dabei sind auf die vorangegangenen Dokumentationen zu referenzieren (Pflichtenheft) und die dortigen Funktions-IDs zu verwenden!

Beispiel:

- F20
- Q10

2.3 Nicht zu testende Funktionen

(optional; auszufüllen, falls es Funktionen gibt, die nicht getestet werden sollen)

Hier werden alle Eigenschaften bzw. Funktionen und Funktionskombinationen aufgelistet, die nicht getestet werden. **Es sollte begründet werden, warum diese nicht getestet werden.** Es versteht sich von selber, dass alle Muss-Funktionalitäten des Pflichtenheftes getestet werden müssen.

2.4 Vorgehen

Die allgemeinen Vorgehensweisen für die einzelnen zu testenden Funktionen und Funktionskombinationen werden hier beschrieben. Die Beschreibung sollte detailliert genug sein, um die Hauptaktivitäten und deren Zeitbedarf abschätzen zu können.

Es ist zu beachten, dass für alle wichtigen Funktionalitäten das Verfahren angegeben wird. Dies gewährleistet, dass diese Funktionalitäten adäquat getestet werden.

Es ist zu dokumentieren, welche Aktivitäten, Techniken und Werkzeuge benötigt werden, damit die Funktionalitäten getestet werden können.

Beispiel für Vorgehen (unvollständige Liste):

a) Abnahme- und Funktionstests

Die Anwendungsfälle aus der Anforderungsspezifikation werden über das Web-Interface geprüft. Mindestanforderung hierfür ist es, jeden Fall einmal auf seine korrekte Funktionalität zu testen.

b) Komponenten- und Integrationstests

Klassen werden mit JUnit-Testfällen geprüft. Vor Beginn der Implementierung werden bereits Blackbox-Testfälle erstellt, die dann begleitend zur Implementierung genutzt werden („Test first“). Nach Abschluss der Implementierung einer Komponente wird diese dann durch Whitebox-Tests geprüft.

Der Integrationstest der Klassen und Komponenten erfolgt nach dem Bottom-Up-Prinzip. Anfangs muss die Integration der Datenbankbindung und den entsprechenden Data-Access-Objects (DAO) geprüft werden, da das Mapping der Datenbank auf Objekte die unterste Schicht

des Projektes bildet. Dieser Testabschnitt wird durch die Schnittstellentests abgedeckt. Die Komponenten werden damit unter Berücksichtigung ihrer Abhängigkeiten konkret in folgender Reihenfolge integriert: ...

(Hier kommt das konkrete Vorgehen bei der Integration: Welche Klassen werden zusammen getestet, welche kommen dann dazu etc. Das kann man z.B. auch schön in Form eines Baumes aufzeigen.)

c) ...

Das Kapitel wird im Laufe des Projekt ergänzt. In der ersten Iteration ist vor allem auf die Abnahmetests einzugehen.

2.5 Testumgebung

Die genutzte Testumgebung(en) bitte hier angeben und kurz beschreiben.

Beispiel: JUnit Testsuite, lokal installierter Web Application Server, ...

Bei Abnahmetest sollte die Testumgebung der Produktumgebung entsprechen.

3 Abnahmetest

Hier wird das Testziel beschrieben.

Dieses Kapitel kann aus der Abnahmetestspezifikation übernommen werden, sollte jedoch die Bearbeitung der Annotationen beinhalten.

Beispiel:

Abnahme des Produkts durch Kunden.

3.1 Zu testende Anforderungen

In diesem Abschnitt werden alle zu testenden Funktionen aufgeführt

Nr	Anforderung	Testfälle	Kommentar
1	<Ref. auf Anforderung> Anforderung	<Referenz auf Testfall>	Hier könnte ein Kommentar stehen
2	<Ref. auf Anforderung> Anforderung	<Referenz auf Testfall>	oder eben auch nicht.

3.2 Testverfahren

Hier wird kurz das Testverfahren beschrieben.

3.2.1 Testskripte

Werden Testskripte (Unittest-Klassen, automatisierte Testsuiten, etc.) verwendet, so sind diese hier zu beschreiben. Wo liegen diese Testskripte und wie sind sie auszuführen.

3.3 Testfälle

Im Folgenden sind so viele Unterkapitel einzufügen, wie es Testfälle gibt.

Jeder Testfall erhält eine eindeutige Identifikation mit Kurzbezeichnung.

Beispiel:

3.3.1 Testfall $\langle T100 \rangle$ - Lager anlegen

Ziel

Zweck des Tests, wie z.B.: Überprüfung des Normalfalls, Überprüfung der Sicherheit

Objekte/Methoden/Funktionen

Hier sind alle Testobjekte und Methoden zu beschreiben, die von diesem Testfall ausgeführt werden. Testobjekte können dabei z.B. auch Komponenten oder einzelne Webseiten sein. Bei Black-Box Tests werden hier die betroffenen Funktionen angegeben.

Beispiel: **F20**

Pass/Fail Kriterien

Es sind die Kriterien anzugeben, mit denen man feststellt, dass der Testfall erfolgreich bzw. fehlgeschlagen ist.

Beispiel:

Überprüfung der Ausgabe durch Testperson. Test erfolgreich, wenn Ausgabe korrekt.

Vorbedingung

Vorbedingungen die erfüllt sein müssen, damit der Test durchgeführt werden kann.

Beispiel:

Im System sind mehr als 100 Nutzer vorhanden

Der Nutzer ist angemeldet

Einzelschritte

Es ist zu beschreiben, was zu tun ist, um einen Testlauf vorzubereiten und ihn zu starten. Ggf. sind erforderliche Schritte während seiner Ausführung anzugeben (z.B. Benutzerinteraktion über ein User-Interface). Ferner ist zu beschreiben, was zu tun ist, um den Testlauf ordnungsgemäß oder im Falle unvorhergesehener Ereignisse anzuhalten (falls er nicht von selbst terminiert). Ggf. sind Aufräumarbeiten zu beschreiben, um nach den Tests den ursprünglichen Zustand wiederherzustellen (falls der Testlauf nicht seiteneffektfrei ist)

Es sind für jeden Schritt die Eingaben und erwartete Ausgaben aufzuführen.

Beispiel:

Eingabe:

1. In Suchfeld klicken
2. Suchbegriff eingeben
3. Mit Enter bestätigen

Ausgabe: Liste von Artikeln entsprechend des Suchbegriffs

Beobachtungen / Log / Umgebung

Es sind alle speziellen Methoden oder Formate zu beschreiben, mit denen die Ergebnisse der Testläufe, die Zwischenfälle und sonstige wichtige Ereignisse aufgenommen werden sollen. Beispiel: Logdatei eines Servers, Messung der Antwortzeit eines Remote Terminals mittels Netzwerk Simulator, ...

Besonderheiten

optional; auszufüllen, falls es Besonderheiten in diesem Testfall gibt.

Testfallspezifische Besonderheiten, z.B. Ausführungsvorschriften oder Abweichungen von der Testumgebung (siehe 2.5) werden hier aufgelistet.

Abhängigkeiten

optional; auszufüllen, falls es Abhängigkeiten in diesem Testfall gibt.

Ist dieser Testfall von der Ausführung anderer Testfälle abhängig, so werden diese Testfälle hier aufgelistet und kurz beschrieben, worin die Abhängigkeit besteht.

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

3.3.2 Testfall $\langle T200 \rangle$ - Lager löschen

Ziel

Objekte/Methoden/Funktionen

Pass/Fail Kriterien

Vorbedingung

Einzelschritte

Beobachtungen / Log / Umgebung

Besonderheiten

Abhängigkeiten

4 Integrationstest

In diesem Kapitel soll die Vorgehensweise beim Integrationstest spezifiziert werden. Dazu wird hier zunächst auf das Testziel eingegangen und in den weiteren Abschnitten auf die zu testenden Komponenten, das Testverfahren, die Testskripte und am Ende die konkreten Testfälle. Die Bearbeitung soll genauso erfolgen, wie bereits in der Abnahmetestspezifikation der Abnahmetest. Achtet bitte darauf, dass die Nummerierung der Testfälle sich nicht mit der aus der Abnahmetestspezifikation überschneidet.

Testziel...

4.1 Zu testende Komponenten

Nr	Komponenten	Testfälle	Kommentar
1	<Ref. auf Komponente> Name der Komponente, ...	<Referenz auf Testfall>	Kommentar
2	<Ref. auf Komponente> Name der Komponente, ...	<Referenz auf Testfall>	

4.2 Testverfahren

...

4.2.1 Testskripte

...

4.3 Testfälle

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

4.3.1 Testfall $\langle T_{200} \rangle$ - Komponente x+y

Ziel

Objekte/Methoden/Funktionen

Pass/Fail Kriterien

Vorbedingung

Einzelschritte

Beobachtungen / Log / Umgebung

Besonderheiten

Abhängigkeiten

5 Unit-Tests

In diesem Kapitel soll die Vorgehensweise im Unit-Test beschrieben werden. Die Bearbeitung soll genauso erfolgen, wie bereits in der Abnahmetestspezifikation der Abnahmetest. Achtet bitte darauf, dass die Nummerierung der Testfälle sich nicht mit der aus der Abnahmetestspezifikation überschneidet.

Testziel...

5.1 Zu testende Komponenten

Nr	Komponenten	Testfälle	Kommentar
1	<Ref. auf Komponente> Name der Komponente	<Referenz auf Testfall>	Kommentar
2	<Ref. auf Komponente> Name der Komponente	<Referenz auf Testfall>	

5.2 Testverfahren

...

5.2.1 Testskripte

...

5.3 Testfälle

<TITEL DES PRAKTIKUMS>

<Titel des Teilprojektes>

5.3.1 Testfall $\langle T300 \rangle$ - Klasse x

Ziel

Objekte/Methoden/Funktionen

Pass/Fail Kriterien

Vorbedingung

Einzelschritte

Beobachtungen / Log / Umgebung

Besonderheiten

Abhängigkeiten

6 Glossar

Hier werden Fachbegriffe erklärt.