

<Titel des Praktikums>

<Titel des Teilprojektes>

Softwareentwicklungspraktikum
Sommersemester 2007

Testdokumentation



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

Betreuer: <Name>

Auftragnehmer: <überzählige Zeilen löschen>

Name	E - Mail

Braunschweig, <Abgabedatum>

Versionsübersicht

Version	Datum	Autor	Status	Kommentar

Status: akzeptiert oder nicht akzeptiert

Kommentar: hier eintragen, was geändert bzw. ergänzt werden musste

Hinweis zum Template:

Dieses Template enthält Hinweise, die alle kursiv geschrieben sind. Alles Kursivgeschriebene ist selbstverständlich bei Abgabe zu entfernen sind.

Angaben in <...> sind mit dem entsprechendem Text zu füllen.

Inhaltsverzeichnis

1	<u>TESTPLAN</u>	5
1.1	ZU TESTENDE KOMPONENTEN	5
1.2	ZU TESTENDE FUNKTIONALITÄTEN	5
1.3	NICHT ZU TESTENDE FUNKTIONALITÄTEN (OPTIONAL; AUSZUFÜLLEN, WO SINNVOLL)	5
1.4	VORGEHEN	5
1.5	KRITERIEN FÜR ERFOLGREICHE BZW. FEHLGESCHLAGENE TESTLÄUFE	6
1.6	TESTUMGEBUNG	6
2	<u>TESTSKRIPT</u>	7
2.1	TESTSKRIPT ID	7
2.1.1	TESTZIEL	7
2.1.2	SPEZIELLE ANFORDERUNGEN	7
2.1.3	TESTABLAUF	7
3	<u>TESTFALL</u>	9
3.1	TESTFALL ID	9
3.1.1	ZU TESTENDE METHODEN	9
3.1.2	EINGABEN	9
3.1.3	AUSGABEN	9
3.1.4	TESTUMGEBUNG	9
3.1.5	BESONDERHEITEN (OPTIONAL; AUSZUFÜLLEN, WO SINNVOLL)	9
3.1.6	ABHÄNGIGKEITEN (OPTIONAL; AUSZUFÜLLEN, WO SINNVOLL)	10
4	<u>TESTPROTOKOLL</u>	11
4.1	TESTPROTOKOLL ID	11
4.2	BESCHREIBUNG	11
4.3	AUFSTELLUNG DER DURCHGEFÜHRTEN TESTFÄLLE	11
4.3.1	AUSFÜHRUNG	11
4.3.2	ERGEBNIS	11
4.3.3	TESTUMGEBUNG (OPTIONAL)	11
4.3.4	UNERWARTETE EREIGNISSE	12
5	<u>ZUSAMMENFASSUNG</u>	13

1 Testplan

Alle Tätigkeiten für das Testen sollten im Voraus geplant werden, um einen möglichst hohen Nutzen aus den Tests zu erhalten und um sicherzustellen, dass alle wichtigen Funktionalitäten systematisch und möglichst vollständig getestet werden. Nachfolgend ist ein Muster angegeben, nach dem ein Testplan erstellt werden kann.

1.1 Zu testende Komponenten

Sämtliche zu testenden Komponenten werden hier aufgeführt. Sind bereits Fehler bei den zu testenden Objekten bekannt, werden diese hier ebenfalls dokumentiert. Dies gilt auch für Objekte, die nicht getestet werden.

1.2 Zu testende Funktionalitäten

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

Beispiel: /F100/ : Benutzer Login

1.3 Nicht zu testende Funktionalitäten (optional; auszufüllen, wo sinnvoll)

*Hier eventuell angeben, welche Funktionalitäten nicht getestet werden sollen. **Alle Muss-Funktionalitäten des Pflichtenheftes (Abschnitt 1.1) müssen getestet werden.***

1.4 Vorgehen

Hier bitte das allgemeine Vorgehen beim Testen beschreiben.

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.

Beispiele:

- Komponententests

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.

- Integrationstests:

Der Integrationstest der Klassen und Komponenten erfolgt nach dem Bottom-Up-Prinzip. Zuerst wird die Integration der Datenbankbindung durch Schnittstellentests mit Mocks / Dummies getestet. Die darauf aufsetzenden 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.)

- Funktionstests

Die Anwendungsfälle aus der Anforderungsspezifikation werden über das Web-Interface geprüft. Dabei sind der Regelfall (korrekte Eingabe aller erforderlichen Daten durch den Benutzer) und Grenzfälle (fehlerhafte oder gar keine Dateneingabe durch den Benutzer) zu testen.

- ... (Weitere hier nicht aufgeführte Vorgehensweisen beim Testen sind projektspezifisch möglich).

1.5 Kriterien für erfolgreiche bzw. fehlgeschlagene Testläufe

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

1.6 Testumgebung

Die genutzte Testumgebung(en) bitte hier angeben.

Beispiel: JUnit Testsuite, lokal installierter Web Application Server und andere

2 Testskript

Das Testskript definiert, wie die einzelnen Testfälle auszuführen sind. Es enthält Informationen über Voraussetzung zum Start der Testfälle und wie der Testfall auszuführen ist.

Ein Testskript sollte für jeweils möglichst viele Testfälle mit ähnlichen Voraussetzungen und ähnlichem Ablauf gültig sein. Für abweichende Gruppen von Testfällen sind jeweils separate Testskripte anzugeben. Testskripte, die nur einen einzigen Testfall enthalten, sind wo möglich zu vermeiden.

2.1 Testskript ID

Werden mehrere Testskripte benötigt, so erhält jedes eine eindeutige Identifikation. Es sind so viele Abschnitte einzufügen, wie es Testskripte gibt.

2.1.1 Testziel

Sinn und Zweck des Testskriptes werden hier beschrieben. Außerdem ist eine Liste aller Testfälle **mit Referenz auf die genaue Testfall-ID** (Abschnitt 3.x), für die dieses Testskript gültig ist.

Beispiel:

- Reaktion auf Benutzereingaben: /T100/, /T300/, /T500/
- Datenbankzugriff: /T200/, /T400/

2.1.2 Spezielle Anforderungen

Besondere Anforderungen, die für die Ausführung des Testskriptes erforderlich sind, werden hier aufgelistet, z.B. Rahmenbedingungen des Systemzustands.

Beispiel: Benutzer ist eingeloggt

2.1.3 Testablauf

Die einzelnen Schritte jedes Testfalls in diesem Testskript sollten im Vorfeld feststehen, um z. B. schon vor Testfallausführung zu klären, was geschehen soll, wenn etwas schief geht. Die Beschreibung des Testablaufs sollte folgende Elemente enthalten:

- Log: Es sind alle speziellen Methoden oder Formate zu beschreiben, um die Ergebnisse der Testläufe, die Zwischenfälle und sonstige wichtige Ereignisse aufzuzeichnen, z.B. Log-Datei im XML-Format.
- Vorbereitungen (optional): Es ist zu beschreiben, was zu tun ist, um die Ausführung des Testskripts vorzubereiten.

- *Start: Es ist zu beschreiben, was zu tun ist, um das Testskript zu starten.*
- *Ausführung (optional): Es ist zu beschreiben, was während der Ausführung des Testskripts zu tun ist, z.B. weitere Benutzereingaben.*
- *Beobachtung: Es ist zu beschreiben, wie Messungen und Testergebnisse gewonnen werden sollen (z.B. die Antwortzeit eines Remote Terminals ist mit einem Netzwerk Simulator zu testen).*
- *Abbruch (optional; auszufüllen, wo sinnvoll): Es ist zu beschreiben, was zu tun ist, um wegen unvorhergesehenen Ereignissen den Test abubrechen.*
- *Neustart (optional; auszufüllen, wo sinnvoll): Es sind alle Startpunkte und das jeweilige Vorgehen zu beschreiben, an denen der Test wieder aufgenommen werden kann.*
- *Stopp (optional; auszufüllen, wo sinnvoll): Es ist zu beschreiben, was zu tun ist, um das Testskript ordnungsgemäß anzuhalten.*
- *Aufräumen (optional; auszufüllen, wo sinnvoll): Es sind die Aufräumarbeiten zu beschreiben, um nach den Tests den ursprünglichen Zustand wiederherzustellen.*

3 Testfall

Ein Testfall ist eine Kombination von Eingabedaten, Bedingungen und erwarteten Ausgaben, die einem bestimmten Zweck erfüllen. Man prüft z.B., ob Vorgaben in einem Spezifikationsdokument eingehalten werden oder ob der Programmablauf tatsächlich dem erwarteten Pfad entspricht.

Der Testfall enthält keine Angaben zur Bedienung oder Ausführung des Testfalls. Diese findet man immer im dazugehörigen Testskript, das auf die Testfälle verweist.

3.1 Testfall ID

Jeder Testfall erhält eine eindeutige Identifikation mit Kurzbezeichnung. Es sind so viele Abschnitte einzufügen, wie es Testfälle gibt

Beispiel: /T100/ Lager anlegen

3.1.1 Zu testende Methoden

Hier sind alle Testobjekte und Methoden zu beschreiben, die von diesem Testfall ausgeführt werden. Testobjekte können dabei z.B. separate Softwaremodule oder einzelne Webseiten sein.

3.1.2 Eingaben

Es sind alle Eingabedaten aufzuführen, die für die Ausführung des Testfalls notwendig sind.

Diese können sowohl als Wert angegeben werden (ggf. mit Toleranzen) als auch als Name, falls es sich um konstante Tabellen oder um Dateien handelt. Außerdem sind alle betroffenen Datenbanken, Dateien, Terminal Meldungen, etc. anzugeben.

3.1.3 Ausgaben

*Es sind alle Ausgaben anzugeben, die erwartet werden (**Soll-Reaktion**).*

3.1.4 Testumgebung

Die genutzte Testumgebung wird hier angegeben. Es ist darauf zu achten, dass alle erwähnten Testumgebungen auch in Abschnitt 1.6 auftauchen.

3.1.5 Besonderheiten (optional; auszufüllen, wo sinnvoll)

Testfallspezifische Besonderheiten, z. B. Ausführungsvorschriften oder besondere Rahmenbedingungen, die nicht in Abschnitt 2.x.2 erwähnt wurden, werden hier aufgelistet.

3.1.6 Abhängigkeiten (optional; auszufüllen, wo sinnvoll)

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

4 Testprotokoll

Das Testprotokoll enthält Angaben über alle ausgeführten Testfälle, deren Ergebnisse und evtl. Abweichungen vom erwarteten Ergebnis.

4.1 Testprotokoll ID

Jedes Testprotokoll erhält eine eindeutige Identifikation. Es sind so viele Abschnitte einzufügen, wie es Testprotokolle gibt

4.2 Beschreibung

Die Testdurchführung wird hier beschrieben. Dabei ist auf folgende Einzelheiten einzugehen:

- *Die getesteten Softwarekomponenten*
- *Testumgebung. Alle wichtigen Parameter sollten beschrieben werden, wie das verwendete Betriebssystem, sonstige mitverwendete Software wie Datenbank, Webserver, Application Server, alle erforderlichen Konfigurationsdetails usw.*

4.3 Aufstellung der durchgeführten Testfälle

Die Ausführung jedes Testfalls muss dokumentiert werden. Die nachfolgende Gliederung ist für alle Testdurchläufe zu übernehmen.

4.3.1 Ausführung

- *Testskript ID*
- *Testfall ID*
- *Kurzbeschreibung*

4.3.2 Ergebnis

Für jeden Testlauf werden die sichtbaren Ergebnisse aufgelistet (z.B. Fehlermeldungen, Programmabbrüche und Meldungen, die von den Testern eine Aktion verlangten).

Natürlich ist auch zu vermerken, ob der Test erfolgreich durchgeführt werden konnte oder nicht.

4.3.3 Testumgebung (optional)

Falls Abweichung zu vorhergehenden Angaben (Abschnitt 3.x.4) vorhanden sind, so werden diese hier aufgeführt.

4.3.4 Unerwartete Ereignisse

Hier wird beschrieben, was vor und nach den unerwarteten Ereignissen passiert ist. Falls Testfälle nicht ausgeführt werden konnten, werden die Umstände beschrieben, die zu diesem Problem geführt haben. Auch wenn keine unerwarteten Ereignisse auftreten, ist dies zu vermerken.

5 Zusammenfassung

Hier wird eine Zusammenfassung der Testergebnisse aufgelistet. Dabei sind alle Probleme, die behandelt wurden und wie ihre Lösungen erreicht wurden aufzulisten.

Dabei kann auf folgendes eingegangen werden:

- *Umfang der Testverlaufs (Vollständigkeit) im Vergleich zu Umfangskriterien des Testplans. Auflistung der Funktionalitäten, die nicht getestet wurden. Selbstverständlich mit Begründung.*
- *Ggf. Abweichungen der durchgeführten Testes vom Testplan, Testskript oder Testfällen. Die Abweichungen sollten begründet werden.*
- *Zusammenfassung der Ergebnisse der Tests, Bewertung der Softwarequalität*
- *Aktivitäten (optional): Zusammenfassung aller Hauptaktivitäten und der wichtigen Ereignisse der Tests*