

Software-Projekt 2 2014

VAK 03-BA-901.02

Testplan



Patrick Hollatz	phollatz@tzi.de	2596537
Tobias Dellert	tode@tzi.de	2936941
Tim Ellhoff	tellhoff@tzi.de	2520913
Daniel Pupat	dpupat@tzi.de	2703053
Olga Miloevich	halfelv@tzi.de	2586817
Tim Wiechers	tim3@tzi.de	2925222

Inhaltsverzeichnis

1	Einführung	3
1.1	Zweck	3
1.2	Umfang	3
1.3	Beziehungen zu anderen Dokumenten	3
1.4	Aufbau der Testbezeichner	3
1.5	Dokumentation der Testergebnisse	3
1.6	Definitionen und Akronyme	4
1.7	Referenzen	4
2	Systemüberblick	4
2.1	Module der Anwendungsschicht und deren Funktionen	4
3	Merkmale	5
3.1	Zu testende Merkmale	5
3.1.1	Funktionale Anforderungen	5
3.2	Nicht zu testende Merkmale	5
4	Abnahme- und Testendekriterien	5
5	Vorgehensweise	5
5.1	Komponenten- und Integrationstest	5
5.2	Funktionstest	5
6	Aufhebung und Wiederaufnahme	5
7	Hardware- und Softwareanforderungen	5
7.1	Hardware	6
7.2	Software	6
8	Testfälle	6
8.1	Komponententest	6
8.2	Integrationstest	7
8.3	Funktionstest	7
8.4	Leistungstest	7
8.4.1	Härtetest	8
8.4.2	Volumentest	8
8.4.3	Sicherheitstest	8
8.4.4	Erholungstest	8
9	Testzeitplan	8

Version und Änderungsgeschichte

Die aktuelle Versionsnummer des Dokumentes sollte eindeutig und gut zu identifizieren sein, hier und optimalerweise auf dem Titelblatt.

Version	Datum	Änderungen
1.0	TT.MM.JJJJ	Dokumentvorlage als initiale Fassung kopiert
1.1	TT.MM.JJJJ

1 Einführung

1.1 Zweck

Was ist der Zweck dieses Testplans? Wer sind die LeserInnen?

1.2 Umfang

1.3 Beziehungen zu anderen Dokumenten

1.4 Aufbau der Testbezeichner

Jeder Testfall muss einen eindeutigen Bezeichner haben. Der könnte hierarchisch aufgebaut sein. Hier wird dieses Namensschema einheitlich festgelegt.

1.5 Dokumentation der Testergebnisse

In welcher Form werden die Testergebnisse dokumentiert (\Rightarrow Testprotokoll)? Wie werden gefundene Fehler dokumentiert (Bugtracking)?

1.6 Definitionen und Akronyme

1.7 Referenzen

2 Systemüberblick

Dieser Abschnitt soll einen Überblick über das System geben, und zwar insbesondere hinsichtlich jener Komponenten, die durch Komponententests geprüft werden sollen. Dazu ist die richtige Granularität der Komponenten wichtig (einzelne Klassen, ganze Pakete oder Systeme etc.). Für den Integrationstest sind die Abhängigkeiten der Komponenten voneinander von besonderer Wichtigkeit.

Der Überblick sollte leicht verständlich sein und sich auf die wesentlichen benötigten Informationen konzentrieren. Hier sind evtl. auch Referenzen auf die Architekturbeschreibung angebracht.

Es sollten die Hofmeister-Konzepte verwendet werden. Benutzt am besten Diagramme aus der Architekturbeschreibung.

Dieser Abschnitt ist nicht einfach nur Copy&Paste der Architekturbeschreibung. Es gilt einerseits, Redundanzen soweit wie möglich zu vermeiden, und andererseits, dieses Dokument so selbsterklärend wie möglich zu machen.

2.1 Module der Anwendungsschicht und deren Funktionen

Hier solltet ihr dann die einzelnen Module aus Sicht des Tests weiter verfeinern. Auch wäre noch eine Grafik angebracht, die die Abhängigkeiten der Subsysteme/Module zeigt, sowie eine kurze Beschreibung dazu.

3 Merkmale

Hier sollen die zu testenden Merkmale, aber auch die nicht zu testenden Merkmale (mit Begründung) aufgelistet werden. Welche Kombinationen von Merkmalen sind relevant?

3.1 Zu testende Merkmale

3.1.1 Funktionale Anforderungen

Welche sind besonders wichtig?

3.2 Nicht zu testende Merkmale

4 Abnahme- und Testendekriterien

Wann wird das Testen beendet? Die angegebenen Kriterien müssen objektiv prüfbar sein.

5 Vorgehensweise

5.1 Komponenten- und Integrationstest

Hier findet sich das konkrete Vorgehen bei der Integration: Welche Klassen werden zunächst zusammen getestet, welche kommen dann hinzu? Das kann man z.B. geeignet in Form eines Baumes aufzeigen.

5.2 Funktionstest

6 Aufhebung und Wiederaufnahme

7 Hardware- und Softwareanforderungen

Hier soll das zu prüfende Material spezifiziert werden. Auf welchen Rechnern werden die Tests durchgeführt? Welche anderen Hilfsmittel oder Ressourcen sind dafür nötig?

7.1 Hardware

7.2 Software

8 Testfälle

*Dies ist der wichtigste (und vermutlich umfangreichste) Teil des Testplans. Hier wird genau aufgelistet, **was wie** und **von wem** getestet wird. Das spätere Testen besteht dann einfach aus einer Durchführung dieser Tests.*

*Zu jedem Testfall muss es eine **Testfallspezifikation** geben (außer Komponententests, siehe unten). Schwerpunkt sollen hier Integrations- und Leistungstests sein. Welche Arten von Integrationstests und Leistungstests seht Ihr vor und wie wollt Ihr diese genau ausgestalten?*

8.1 Komponententest

Auf die genaue Spezifikation der Komponententests könnt Ihr verzichten – diese werden durch die JUnit-Testfälle gegeben. Uns genügt hierzu eine Beschreibung, welche Komponenten/Klassen wie, wann und durch wen getestet werden sollen. Die zugehörigen JUnit-Tests werden separat abgegeben (Black-Box jetzt, White-Box später bei der Abgabe der Implementierung) und müssen im Testplan nicht als Code aufgeführt werden.

Klasse	Implementierer	Tester	Testart
ReferenceManagement	Klaus	Hans	Blackbox
UserManagement	Anna	Klaus	Blackbox
...
ReferenceTable	Heinz	Bert	Whitebox

Tabelle 1: Komponententests

Achtet darauf, dass der Tester ein anderer ist als der Implementierer. Jeder soll mindestens eine Klasse als Black-Box und eine Klasse als White-Box testen!

Hier könnten später noch die Kontrollflussdiagramme für die Whiteboxtests eingefügt werden.

8.2 Integrationstest

Die Integrationstests werden hier genauer beschrieben: Welche Klassen sind beteiligt? Wie ist der Zustand des Systems vor Beginn? Welche Eingaben werden getätigt? Welche Ergebnisse/welches Verhalten wird erwartet?

Aufbau einer Testfallspezifikation:

- 1. eindeutiger Testfallbezeichner (entspricht Namenskonvention aus Abschnitt 1.4)*
- 2. Testobjekte: welche Komponenten werden getestet?*
- 3. Eingabespezifikationen (Eingaben des Testfalls)*
- 4. Ausgabespezifikationen (erwartete Ausgaben)*
- 5. Umgebungserfordernisse (notwendige Software- und Hardwareplattform sowie Testtreiber und -rumpfe)*
- 6. besondere prozedurale Anforderungen (Einschränkungen wie Zeitvorgaben, Belastung oder Eingreifen durch den Operator)*
- 7. Abhängigkeiten zwischen Testfällen*

8.3 Funktionstest

Die Funktionstests basieren auf den Anwendungsfällen. Diese müssen hier nun konkretisiert werden, um einen Testfall zu erhalten. Es müssen also konkrete Werte für Ein- und Ausgabe festgelegt werden. Auch das Verhalten im Fehlerfall sollte getestet werden.

Die Anwendungsfälle müssen nicht alle wiederholt werden. Sie werden schließlich in der Anforderungsspezifikation spezifiziert. Hier genügt eine tabellarische Auflistung aller zu testenden Anwendungsfälle, deren konkrete Ein- und Ausgaben sowie die Umsetzung (automatisiert oder manuell und falls manuell, durch welche Art von Benutzer?). Insbesondere müsst Ihr an dieser Stelle klären, welche Varianten von Anwendungsfällen getestet werden. Falls Varianten oder gar ganze Anwendungsfälle nicht getestet werden sollen, dann wird hier eine plausible Begründung erwartet.

8.4 Leistungstest

Die Leistungstests prüfen die Leistungsanforderungen, wie z.B. Reaktionszeiten, das Verhalten unter extremen Bedingungen, bei großen Datenmengen etc. . .

8.4.1 Härtetest

8.4.2 Volumentest

8.4.3 Sicherheitstest

8.4.4 Erholungstest

Hier wird geprüft, ob sich das System von Fehlerzuständen auch wieder erholt. Es werden also gezielt Fehler provoziert, um die Korrektheit der Systemreaktion herauszufinden.

9 Testzeitplan