

Online-Library Master Test Plan

Version 1.0

Revision History

Date	Version	Description	Author
18.04.2014	1.0	Master Test Plan	Torsten, Wanping

Table of Contents

Inhaltsverzeichnis

1.	Introduction.....	4
1.1	Purpose.....	4
1.2	Scope.....	4
1.3	Intended Audience.....	4
1.4	Document Terminology and Acronyms.....	4
1.5	References	4
1.6	Document Structure	4
2.	Evaluation Mission and Test Motivation	4
2.1	Background	4
2.2	Evaluation Mission.....	4
2.3	Test Motivators.....	4
3.	Target Test Items	4
4.	Test Approach	5
4.1	Initial Test-Idea Catalogs and Other Reference Sources	5
4.2	Testing Techniques and Types	5
4.2.1	State-based Testing (Server)	5
4.2.2	Business Cycle Testing (Server).....	5
4.2.3	User Interface Testing (Client)	5
5.	Entry and Exit Criteria.....	6
5.1	Test Plan.....	6
5.1.1	Test Plan Entry Criteria	6
5.1.2	Test Plan Exit Criteria	6
5.2	Test Cycles	6
5.2.1	Test Cycle Entry Criteria	6
6.	Deliverables	6
6.1	Test Evaluation Summaries.....	6
6.2	Reporting on Test Coverage	6
7.	Test Data Management	6

Master Test Plan

1. Introduction

1.1 Purpose

Der Zweck des Iteration Test Plan ist es, alle notwendigen Informationen zur Planung und Steuerung des Testaufwands für eine bestimmte Iteration zu sammeln. Es beschreibt den Ansatz für die Prüfung der Software und ist der Top-Level-Plan generiert und genutzt von Managern, den Testaufwand zu lenken.

Dieser Testplan für Necabatur unterstützt die folgenden Ziele:

- Automatisierte Server-Tests
- Manuelle Client-Tests
- Test-Vorgehensweise
- Test-Ressourcen und Aufwände

1.2 Scope

- State-based Tests (Server)
- Business Cycle Tests (Server)
- User-Interface-Tests (Client)
- Last-test (System)

1.3 Intended Audience

- Project Manager
- Project Members

1.4 Document Terminology and Acronyms

1.5 References

1.6 Document Structure

2. Evaluation Mission and Test Motivation

2.1 Background

- Funktionieren der Server-Logik sicherstellen

2.2 Evaluation Mission

- Fehler in der Logik auffinden
- Ausreichende Performance, bei Anfragen und Suchen, aber auch beim Eintragen, sicherstellen

2.3 Test Motivators

- Use-Cases
- Performance
- Ablauf

3. Target Test Items

- Client
- Server
- System

4. Test Approach

4.1 Initial Test-Idea Catalogs and Other Reference Sources

4.2 Testing Techniques and Types

4.2.1 State-based Testing (Server)

Technique Objective:	Funktionierende Zustandsübergänge sicherstellen (Server)
Technique:	Basierend auf Unit-Tests werden Transfer-Objekte an den Server gesendet und die spezifizierten Antworten zurückerwartet.
Oracles:	Ergebnis der Unit-Tests und das Test-Log
Required Tools:	Visual Studio Unit Test
Success Criteria:	Alle Tests für den spezifizierten Zustandsübergang müssen erfolgreich sein (grün).
Special Considerations:	Die Tests müssen Anfragen und Antworten gemäß der Kommunikations-Spezifikation versenden bzw. erwarten.

4.2.2 Business Cycle Testing (Server)

Technique Objective:	Funktionierende Logik/Ablauf sicherstellen (Server)
Technique:	Basierend auf Unit-Tests werden Transfer-Objekte an den Server gesendet und die spezifizierten Antworten zurückerwartet. Durch die Verzahnung der einzelnen Tests wird ein ganzer Business Cycle (Vorgang: Ausleihen, Neue Sachen hochladen) abgedeckt.
Oracles:	Ergebnis der Unit-Tests und das Test-Log
Required Tools:	Visual Studio Unit Test
Success Criteria:	Alle Tests für den spezifizierten Zustandsübergang müssen erfolgreich sein (grün).
Special Considerations:	Die Tests müssen Anfragen und Antworten gemäß der Kommunikations-Spezifikation versenden bzw. erwarten.

4.2.3 User Interface Testing (Client)

Technique Objective:	Funktionalität der grafischen Nutzeroberfläche gewährleisten
Technique:	Nutzeraktionen (Mausklick, Tastatureingaben) werden durch ein Tool emuliert und die danach vorherrschende Oberfläche mit einem definierten Soll-Zustand verglichen.
Oracles:	Der Erfolg des Tests ist gegeben, wenn die grafische Oberfläche einen bestimmten Zustand eingenommen hat. Der Eintritt dieses Zustands kann festgestellt werden, indem man automatisiert auf die Existenz von Oberflächen-Komponenten (Schaltfläche, Eingabefehler, Bilder...) prüft.
Required Tools:	Visual Studio
Success Criteria:	Alle Oberflächen-Tests müssen erfolgreich sein (grün).
Special Considerations:	Teile des Clients sind nur in einem laufenden Vorgang erreichbar. Um ein

	Vorgang starten zu können, sind mehrere Clients nötig. Da die Koordination mehrere mit den vorhandenen Tools nicht möglich ist, können nur die Client-Bestandteile getestet werden, die ohne laufenden Vorgang erreichbar sind.
--	---

5. Entry and Exit Criteria

5.1 Test Plan

5.1.1 Test Plan Entry Criteria

Mit Implementierung der ersten Funktionalität können die Tests beginnen.

5.1.2 Test Plan Exit Criteria

Der Testplan ist durch den Erfolg der manuellen Tests beendet.

5.2 Test Cycles

5.2.1 Test Cycle Entry Criteria

Die Erfüllung der Spezifikation durch die Tests und die Implementierung muss überprüft werden und dann die Tests nochmals ausgeführt werden.

6. Deliverables

6.1 Test Evaluation Summaries

Test-Logs

6.2 Reporting on Test Coverage

Coverage-Logs

7. Test Data Management

Testfälle benötigen für die Ausführung meist Daten, die bestimmte Rahmenbedingungen der Testausführung festlegen. In den meisten Fällen ist es sinnvoll denselben Testfall mehrfach mit unterschiedlichen Daten auszuführen, um sogenannte „falsch negative“ Ergebnisse zu vermeiden. Zudem ist es oft wünschenswert, die Daten später mit geringerem Aufwand ändern zu können, weil bei der Testerstellung nicht immer alle Fälle bedacht werden können.

Deshalb werden die Konfigurationsdaten für die Tests explizit gespeichert, welche in einem eigenen Unterverzeichnis des Projektes angelegt werden sollen. Für jeden Testfall existiert eine spezielle Konfigurationsdatei, in welcher die verschiedenen Testdurchläufe mit den jeweiligen Parametern definiert sind. Darin können sehr einfach weitere Testdurchläufe ergänzt oder bereits Vorhandene geändert werden. Für jeden definierten Testdurchlauf eines Testfalls wird bei der Test-Ausführung dann automatisch eine eigene Instanz des Testfalls mit der entsprechenden Parametrierung erzeugt und ausgeführt.