Testkonzept

| **Auftraggeber** | LFH |
| --- | --- |
| **Projektleiter** | Andy Villiger |
| **Autor** | Sandro Dallo |
| **Status** | In Arbeit |

Änderungsverzeichnis

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 17.11.2013 | 0.1 | Initialer Entwurf | Sandro Dallo |
| 17.11.2013 | 0.2 | Fehlerklassen, Fehler melden, Test Ende/Abbruch | Jonas Alder |
|  |  |  |  |

# Inhaltsverzeichnis

1 Inhaltsverzeichnis 2

2 Test Übersicht 3

2.1 Testziele 3

2.2 Test Strategie 3

2.3 Testobjekte 3

2.4 Testarten 3

2.5 Testabdeckung 4

3 Testrahmen 5

3.1 Testvoraussetzungen 5

3.2 Fehlerklassen 5

4 Testinfrastruktur 6

4.1 Testsystem 6

4.2 Testdaten 6

4.3 Testhilfsmittel 6

5 Testfallbeschreibungen 7

6 Testplan 8

# Test Übersicht

## Testziele

Das Sicherstellen der geforderten Qualität des Webshop Eshomo gemäss Pflichtenheft[[1]](#footnote-1). Verhindern von folge Fehlern bei neu entwickelten Komponenten, welche dem Gesamtsystem hinzugefügt wurden. Änderungen der Implementierung von bestehenden Komponenten sollen keinen unerwünschten Einfluss auf das Gesamtsystem haben.

Die Tests müssen zu jederzeit nachvollziehbar sein und müssen jeweils mit den gleichen Voraussetzungen durchgeführt werden.

## Test Strategie

Die gewählte Teststrategie basiert auf „Risk based testing“ und berücksichtigt bei der Testabdeckung, die Risiken welche beim Auftreten eines Fehlers auftreten. Somit müssen mindestens die Komponenten durchgängig getestet werden, welche Daten verändern sowie solche die den Verkauf von Produkten massgeblich behindern. Zudem sollen alle definierten Use Cases mit entsprechenden Oberflächentests abgedeckt werden.

## Test Ende

Das Test Ende ist erreicht, wenn alle geplanten Tests durchgeführt wurden.

### Test Abbruch

Die Test Serie wird abgebrochen, falls ein oder mehrere Fehler der Klasse 4 (siehe 3.2) gefunden werden.

## Testobjekte

|  |  |  |
| --- | --- | --- |
| Nr | Objekt | Beschreibung |
| 1 | GUI Layer | Die Benutzeroberfläche des Webshop |
| 2 | Business Layer | Die komplette Businesslogik der Anwendung |
| 3 | Data Access Layer | Der Zugang zu persistenten Daten (DB, Dateien, etc…) |
| 4 | Schnittstellen API | Die verantwortliche Schnittstelle für den automatischen Austausch mit dem MAVE Projekt |

## Testarten

|  |  |  |
| --- | --- | --- |
| Nr | Testart | Beschreibung |
| 1 | Funktionale Tests | Tests um die gewünschten Funktionen der einzelnen Komponenten und Module zu testen. Hauptsächlich mit Hilfe von Unit Tests realisiert. |
| 2 | Integrationstests | Tests über einen ganzen Business Prozess mit möglichst allen Komponenten welcher der Prozess benötigt. Realisiert mit Testframework oder manuell. |
| 3 | Nicht Funktionale Tests | Nicht Funktionale Anforderungen an die Anwendung werden manuell getestet. |
| 4 | Oberflächentests | Tests um die Funktionalität des GUI sowie das Aussehen zu prüfen. Realisiert mit Testframework und manuell. |

## Testabdeckung

Übersicht Testfälle

|  |  |  |
| --- | --- | --- |
| Nr | Testobjekt | Testfälle |
| 1 | GUI Layer | t.b.d |
| 2 | Business Layer | t.b.d |
| 3 | Data Access Layer | t.b.d |
| 4 | Schnittstellen API | t.b.d |

# Testrahmen

## Testvoraussetzungen

Die Tester bestehen aus dem kompletten Entwickler Team und benötigen deshalb keine speziellen Kenntnisse. Die Tests werden laufend während der Realisierung entwickelt und jeweils getestet. Nach der Fertigstellung der Anwendung, werden alle automatischen Test, sowie die manuellen Tests nochmals durchgeführt.

## Fehlerklassen

Festgestellte Fehler werden in Klassen 1-4 eingestuft

|  |  |  |
| --- | --- | --- |
| Nr. | Fehlerklassen | Beschreibung |
| *1* | *Unwesentlicher Mangel* | Mangel wird zur Pendenzenliste hinzugefügt. Es kann ungehindert weiter getestet werden. |
| *2* | *Mangel* | Mangel wird zur Pendenzenliste hinzugefügt. Es kann mit Einschränkungen weiter getestet werden. |
| *3* | *Kritischer Mangel* | Mangel wird zur Pendenzenliste hinzugefügt. Folgende zusammenhängende Tests können nicht durchgeführt werden. |
| *4* | *Blocker* | Mangel wird zur Pendenzenliste hinzugefügt. Es kann nicht weiter getestet werden. Zuständigen Entwickler informieren, muss umgehend korrigiert werden. |

## Fehler melden

Alle gefundenen Fehler müssen im „Case Tracker“ von Open Atrium (von der LFH zur Verfügung gestellten) in der Gruppe „e-Shop“ gemeldet werden[[2]](#footnote-2). Die folgenden Informationen sind dazu anzugeben mindestens:

* Exakte Beschreibung des aufgetretenen Fehlers
* Informationen um den Fehler zu reproduzieren (falls möglich)
* Die Fehlerklasse
* Software auf dem verwendeten Testsystem(en) inkl. Version (Betriebssystem, Browser, Datenbank, Frameworks, Revision/Version des Projekts)

# Testinfrastruktur

## Testsystem

Während der Entwicklung werden die Tests auf den jeweiligen Rechnern der Entwickler durchgeführt. Die Abschlusstests werden auf einer VM [[3]](#footnote-3) welche auch für die Präsentation der Anwendung vorgesehen ist durchgeführt.

Referenz VM

* Ubuntu Server 12.04.3 LTS
* 4 Gb Ram
* 20 Gb Festplattenspeicher
* Play Framework 2.2
* OpenJDK 7 (jdk)
* MySql Server 5.5

## Testdaten

Die Testdaten für Produkte, Kunden und Bestellungen werden von der LFH zur Verfügung gestellt. Genaue Daten können aus dem Anhang entnommen werden.

## Testhilfsmittel

Für die Tests wird das Test Framework von Play verwendet, welches auf JUnit und Selenium basiert.

# Testfallbeschreibungen

|  |  |  |
| --- | --- | --- |
| ID / Bezeichnung | T-001 |  |
| Beschreibung |  | |
| Testvoraussetzung |  | |
| Testschritte |  | |
| Erwartetes Ergebnis |  | |

# Testplan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Aktivität | Verantwortlich | Mitarbeit | Termin |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Pflichtenheft Projekt LFH Version 1.1 [↑](#footnote-ref-1)
2. <http://dev-oa.mave-projekt.de/?q=e-shop/casetracker> [↑](#footnote-ref-2)
3. Virtual Machine – Virtual Box wird verwendet [↑](#footnote-ref-3)