

Übungsblatt 12 (12.01.2016)

Usability Test, Metriken, Use Cases, Testfallableitung

In dieser Übung:

- ✓ Führen Sie einen Usability Test für Ihr *Movie Manager* Produkt durch.
- ✓ Erheben und werten Sie Metriken aus auf dem Quelltext der *Movie Manager* Anwendung und des Webclients.
- ✓ Üben Sie die Bewertung von Entwürfen anhand von Metriken.
- ✓ Üben Sie die Erstellung und Überprüfung von Use Cases für die *Movie Manager* Anwendung
- ✓ Leiten Sie Testfälle aus einem Use Case der *Movie Manager* Anwendung ab.
- ✓ Leiten Sie aus Ihren Erfahrungen Funktionalitäts-Vorschläge für die *Movie Manager* Anwendung ab.

Hinweise zur Weiterentwicklung des *Movie Manager*

Ab diesem Übungsblatt dienen die Aufgaben zum *Movie Manager* zur Verbesserung der Qualität und eventuell auch zur Erweiterung der Funktionalität. Ziel ist es, dass Sie die Weiterentwicklung Ihrer eigenen Lösung üben.

Hinweis Testate ab Übungsblatt 12

Ab diesem Übungsblatt gibt es keine Testataufgaben mehr. Wir empfehlen natürlich, die Aufgaben zu bearbeiten, da die behandelten Themen relevant für die Klausur sind.

Aufgabe 12.1: Usability Test

Präsenz: Ja	Punkte: 8	Team: Ja(2 x 4)	Projekt: Ja	
-------------	-----------	-----------------	-------------	--

Ziel der Aufgabe ist es, teamübergreifend Usability Tests für die *Movie Manager* Anwendung und den *Movie Manager* Webclient durchzuführen. Mit dem Usability Test soll überprüft werden, ob Ihre *Movie Manager* Anwendung oder Ihr *Movie Manager* Webclient hinsichtlich besserer Gebrauchstauglichkeit überarbeitet werden muss. Dazu testen jeweils die zwei Teams, die die Abschlusspräsentation zusammen gehalten haben, gegenseitig ihre Serienverwaltung auf Gebrauchstauglichkeit.

Sie erhalten **Standardanleitungen**, die die Aufgaben der Rollen **TestnutzerIn (VerwalterIn und AusleiherIn)** und **TestbeobachterIn** beim Usability Test beschreiben (siehe Dateien **12-Anleitung-TestnutzerIn.pdf** und **12-Anleitung-TestbeobachterIn.pdf**). Die TestbeobachterInnen erhalten eine Problemlistenvorlage, mit deren Hilfe die gefundenen Beobachtungen zu dokumentieren sind (siehe Datei **12-Problemliste.doc**).

1. Teilen Sie jedes Team so auf, dass 2 Personen als **TestnutzerIn (jeweils eine VerwalterIn und eine AusleiherIn)** fungieren und die anderen Personen als **TestbeobachterIn**. Jedes Team

testet die *Movie Manager* Anwendung und den *Movie Manager* Webclient des jeweils anderen Teams.

2. Lesen Sie sich die **Standardanleitung** für die jeweilige Rolle durch und befolgen Sie diese während des Usability Tests.
3. Entwickeln Sie aus den User Stories der Anforderungen (siehe 05-AnforderungenWebclientAusleihe.pdf) einen Ablauf, in dem alle User Stories abgeprüft werden. Beispielweise kann die VerwalterIn in der *Movie Manager* Anwendung Daten anlegen und diese auf verleiherIn stellen. Die AusleiherIn leiht daraufhin etwas über den Webclient aus und gibt Teile davon zurück. Die VerwalterIn überprüft den Status der Ausleihe in der *Movie Manager* Anwendung. Achten Sie bei Ihrem Ablauf darauf, dass jede User Story mindestens einmal vorkommt. Tragen Sie die User Stories in der Reihenfolge des Ablaufs in die Tabelle in der Problemliste ein.
4. Führen Sie den **Usability Test** durch. Die TestbeobachterInnen protokollieren die Beobachtungen, Probleme und Antworten getrennt in Problemlisten. Protokollieren Sie auch Besonderheiten, die Sie nicht als Problem auffassen. Ordnen Sie allen Problemen eine Fehlerklasse zu. Konsolidieren Sie dann die beiden Listen zu einer Liste. Achten Sie darauf, dass ähnliche Probleme zusammengefasst werden. Schicken Sie die Problemliste möglichst bald an das andere Team.
5. Tragen Sie die vom anderen Team gefundenen Probleme in Ihr UNICASE-Projekt ein. Listen Sie in UNICASE in der Section „Issues“ im „Project Management Document“ die aufgetretenen Probleme (Element „Issue“) auf. Skizzieren Sie für mindestens 4 Issues je zwei Verbesserungsvorschläge (Element „Proposal“).
6. Verbessern Sie anhand der „Proposals“ die aufgetretenen Probleme in Ihrer Implementierung.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.

Bitte speichern Sie Ihre Änderungen am Quellcode auf dem Git Server.

Bitte speichern Sie eine .zip-Datei bis **Montag 18.01.2016 um 10.00 Uhr** in Moodle bestehend aus:

- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)
- 6 exportierten Projekten „movies“, „movies.edit“, „movies.ui“, „movies.web“, „movies.emfstore“ und „movies.web.model“
- Eine konsolidierte Problemliste, die ihr Team erstellt hat als .pdf

Aufgabe 12.2: Erhebung und Auswertung von Metriken

Präsenz: Nein	Punkte: 5	Team: Ja(4)	Projekt: Ja	
---------------	-----------	-------------	-------------	--

Führen Sie auf den von Ihnen neu implementierten bzw. angepassten Klassen zur Umsetzung der Webausleihe eine statische Analyse durch:

1. Benutzen Sie die Funktion „Compute Metrics“ des Eclipse-Plugin „Google CodePro AnalytiX“ und erheben Sie die folgenden Metriken für den von **Ihnen erstellten oder angepassten Quellcode** des *Movie Managers* für die vier Projekte **movies**, **movies.edit**, **movies.ui** und **movies.web**. Unterscheiden Sie für das Projekt **movies.web** die Metriken getrennt nach UI-Code und anderen Code, d.h. bestimmen Sie die Metriken für den UI-Code und für anderen Code in dem Projekt.
 - Anzahl der Quelltextzeilen (Lines of Code)
 - Anzahl der Konstruktoren (Number of Constructors)
 - Anzahl der Attribute (Number of Fields)
 - Anzahl der Methoden (Number of Methods)
 - Anzahl der Typen (Number of Types)

- Durchschnittliche Anzahl Quelltextzeilen pro Operation
 - Durchschnittliche Anzahl Parameter
2. Dokumentieren Sie die Metriken in einer Textdatei. Diese Textdatei soll für die angegebenen Projekte die genannten Metriken beinhalten. **Vergleichen** Sie anschließend diese Metriken mit den unten aufgeführten erhobenen Metriken für den bereits existierenden Code des *Movie Managers*. Vergleichen Sie auch die Metriken der *Movie Manager* Anwendung und des *Movie Manager* Webclients in Bezug auf UI und nicht-UI. Dokumentieren Sie dafür die Gemeinsamkeiten und erklären Sie die Unterschiede in der gleichen Textdatei.

Metrik	movies	movies.edit	movies.ui	movies.web (ui Code)	movies.web (anderer Code)
Anzahl der Quelltextzeilen	8599	1300	9102	0	92
Anzahl der Konstruktoren	15	10	31	0	1
Anzahl der Attribute	290	11	167	0	5
Anzahl der Methoden	445	102	246	0	6
Anzahl der Typen	28	10	101	0	7
Durchschnittliche Anzahl Quelltextzeilen pro Operation	6.69	9.61	8.79	0	6.85
Durchschnittliche Anzahl Parameter	0.61	0.95	0.84	0	1.16

3. Damit Sie Ihre Metriken teamübergreifend vergleichen können, haben wir in Moodle eine Umfrage zu den Metriken eingerichtet. Füllen Sie die Umfrage zu den Metriken bis spätestens Montag, 18.01.2016 in Moodle aus, d.h. beantworten Sie die Fragen indem Sie die berechneten Werte in die Felder der Umfrage eingeben. Die Ergebnisse der Umfrage sind ab Montag, 18.01.2016 um 10:00 Uhr sichtbar.

Ergebnis:

Speichern Sie bitte bis **Montag 18.01.2016 um 10.00 Uhr** in Moodle eine .zip-Datei bestehend aus

- eine PDF-Datei mit den erhobenen Metriken aus 11.1.1 und dem Vergleich aus 11.1.2.

Aufgabe 12.3: Bewertung Dialog vs. Wizard anhand von Metriken

Präsenz: Nein	Punkte: 5	Team: Ja(4)	Projekt: Nein	
---------------	-----------	-------------	---------------	--

Gegeben ist die folgende Beschreibung einer Funktionalität zum Generieren und Exportieren einer Übersicht von verliehenen Serien in der *Movie Manager* Anwendung:

1. Die NutzerIn kann über das Projekt Kontext-Menü eine Aktion auslösen, mit der eine Übersicht über alle verliehenen Serien, den enthaltenen Staffeln und Episoden erzeugt und exportiert werden kann. Bevor die Übersicht generiert wird, erfolgt eine Auflistung aller Serien im Projekt, aus der die NutzerIn die Serien auswählen kann, die in der exportierten Übersicht enthalten sein sollen.
2. Im nächsten Schritt kann die NutzerIn wählen, welche Attribute der Serien, Staffeln und Episoden (Bewertung, Anzahl Schauspieler, ...) in der Übersicht enthalten sein sollen.
3. Im darauffolgenden Schritt kann die NutzerIn eines der Exportformate, Excel-Datei oder CSV, wählen.

4. Im nächsten Schritt, der nochmals alle gewählten Eigenschaften zeigt, wählt die NutzerIn den Speicherort der exportierten Datei. Mit einem Klick auf den Export-Button wird die Übersicht erzeugt und an den angegebenen Speicherort abgelegt.

Weiterhin sind die folgenden zwei Vorschläge zur Umsetzung dieser Funktionalität gegeben:

- a) Verwendung eines mehrstufigen Dialogs, mehrstufig bedeutet, dass nach jedem Schritt (1-4) ein neuer Dialog gestartet und angezeigt wird. Ein Dialog ist ein eigenständiges Fenster auf der GUI. In jedem Schritt wird der jeweilige Dialog über eine eigene Handler-Klasse gestartet. Für jede Handler-Klasse ist dabei zusätzlich eine Command-Klasse notwendig. Als Datenstruktur wird eine Liste von Serien (enthält auch alle Staffeln und Episoden) verwendet. Wird ein neuer Dialog gestartet, so wird auch die Datenstruktur übergeben.
 - b) Als Alternative zum mehrstufigen Dialog kann zur Umsetzung ein Wizard verwendet werden. Ein Wizard ist ein eigenständiges Fenster auf der GUI. Der Wizard umfasst für jeden Schritt (1-4) eine Wizard-Page. Die Wizard-Pages sind Bestandteile eines Wizards, d.h. ein Wizard beinhaltet eine oder mehrere Wizard-Pages, die durch eine festgelegte Reihenfolge nacheinander durchlaufen werden. Zum Starten eines Wizard benötigt man eine Handler-Klasse und eine Command-Klassen. Die einzelnen Wizard-Pages werden nicht gesondert über weitere Handler und Commands aufgerufen. Als Datenstruktur wird wieder eine Liste von Serien (enthält auch alle Staffeln und Episoden) verwendet. Wird ein neuer Wizard gestartet, so wird auch die Datenstruktur übergeben.
1. Schätzen Sie für beide Vorschläge jeweils die folgenden Metriken: Anzahl neu zu erstellender UI-Klassen, Anzahl neuer Handler- und Command-Klassen, Anzahl zu ändernder bereits bestehender Klassen, Anzahl Übergabe der verwendeten Datenstruktur sowie die Anzahl an Mausklicks, die die NutzerIn höchstens benötigt, um die Funktion auszuführen. Gehen Sie davon aus, dass der Speicherort mit einem Klick gewählt werden kann. Begründen Sie Ihre Abschätzung. Speichern Sie die Abschätzung sowie die Begründungen in einer Textdatei.
 2. Bewerten Sie für die beiden Vorschläge anhand der Abschätzung den gesamten Implementierungsaufwand (Anzahl neu zu erstellender UI-Klassen, Anzahl neuer Handler- und Command-Klassen, Anzahl zu ändernder bereits bestehender Klassen, Anzahl Übergabe der verwendeten Datenstruktur) sowie den sich für die NutzerInnen ergebenden Aufwand bei der Ausführung des Dialogs bzw. des Wizards. Nutzen Sie als Bewertungskriterien die folgende Skala:

Skala	Bedeutung
0	wenig Aufwand
1	etwas Aufwand
2	hoher Aufwand

Verwenden Sie die Rationale-Funktionalität von UNICASE, um die Vorschläge gegen die Kriterien (Implementierungsaufwand, NutzerInnenaufwand) zu bewerten: Erstellen Sie ein Element vom Typ „Issue“, in welchem Sie die Vorschläge anhand der Kriterien bewerten und sich für eine Lösung entscheiden. Begründen Sie die Wahl der Lösung. Speichern Sie die Rationale-Elemente im „Design Document“.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.

Bitte speichern Sie eine .zip-Datei bis **Montag 18.01.2016 um 10.00 Uhr** in Moodle bestehend aus:

- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)
- Ermittelte Metriken als .pdf

Aufgabe 12.4: Use Case Beschreibung für die Movie Manager Anwendung

Präsenz: Nein

Punkte: 10

Team: Ja(4)

Projekt: Nein

In dieser Aufgabe üben Sie Use Cases am Beispiel der Nutzungsbeschreibung der *Movie Manager* Anwendung in UNICASE. Die Aufgabe gliedert sich in die nachfolgend beschriebenen zwei Teile:

■ Erstellen eines Nutzungsdiagramm

1. Machen Sie sich mit den Elementen vom Typ Use Case im „ReqSpec Document“ Ihres UNICASE Projekts vertraut. Erstellen Sie ein **Nutzungsdiagramm**, das den Akteur „Manager“, die Aufgabe „Movie Management“, die initiierten Use Cases und Systemfunktionen enthält. Nutzen Sie hierfür den Diagrammtyp Use-Case-Diagram des Modellierungswerkzeugs Astah. Speichern Sie das fertige Nutzungsdiagramm als File Attachment in der Section „Use Cases“ im „ReqSpec Document“ Ihres UNICASE Projekts.

■ Überprüfen und Erstellen von Use Cases

2. Schauen Sie den Use Case **Manage Movie Data** an und versuchen Sie diesen mit dem tatsächlichen Verhalten des Movie Managers nachzuvollziehen. Notieren Sie **2 Unterschiede**, wenn die Beschreibung des Use Cases mit seinen Human Steps/IT System Steps nicht zu den Möglichkeiten passt, die die Benutzeroberfläche anbietet. Notieren Sie die Unterschiede als Elemente vom Typ „Issue“ in dem Abschnitt „Issues“ im „Project Management Document“ in Ihrem UNICASE-Projekt.
3. Erstellen Sie für den Use Case **Manage Ratings** den **Flow of Events** (d.h. Human Steps und IT System Steps) aufbauend auf den Informationen der Vor- und Nachbedingung, den Regeln, der Ausnahmen sowie dem tatsächlichen Verhalten des Movie Managers. Identifizieren Sie die ausgeführten Systemfunktionen und binden Sie diese an richtiger Stelle bei den IT System Steps ein. Beachten Sie bei der Use Case-Beschreibung auch den Stilratgeber für Use Cases in Moodle.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server!

Bitte speichern Sie eine .zip-Datei bis **Montag 18.01.2016 um 10.00 Uhr** in Moodle bestehend aus:

- Nutzungsdiagramm als .asta-Datei und als .png
- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)

Aufgabe 12.5: Ableitung von Testfällen aus Use Cases für die Movie Manager Anwendung

Präsenz: Nein

Punkte: 6

Team: Ja(4)

Projekt: Ja

Ziel dieser Aufgabe ist es, Systemtestfälle aus der Beschreibung eines Use Cases abzuleiten. Gehen Sie dazu wie folgt vor:

1. Leiten Sie logische und konkrete Systemtestfälle für den Use Case **Manage Watched Movies** in Ihrem UNICASE Projekt ab, sodass **alle gültigen und ungültigen Pfade** des Flow-of-Events des Use Cases abgedeckt bzw. geprüft werden (vgl. Foliensatz 10, Folie 87 und 88). Tragen Sie die Testfälle in Ihr UNICASE Projekt in den Abschnitt „Test Document“ unter „System Test“ ein. Verwenden Sie dafür das Systemtest-Modellelement.
2. Führen Sie die Systemtestfälle aus Aufgabe 12.6.1 in der *Movie Manager* Anwendung durch und protokollieren Sie die Ergebnisse des Testfalls in einem Testprotokoll.

3. Vergleichen Sie in einer Textdatei die Ableitung von Testfällen aus Use Cases mit dem Vorgehen zur der Testfallspezifikation aus Aufgabe 7.2 z.B. in Bezug auf Vollständigkeit.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server!

Bitte speichern Sie eine .zip-Datei bis **Montag 18.01.2015 um 10.00 Uhr** in Moodle bestehend aus:

- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)
- Vergleich der Testfallspezifikation als .pdf

Aufgabe 12.6: Vorschlag neuer Funktionalität – Movie Manager Anwendung**Präsenz:** Nein**Punkte:** 6**Team:** Ja(4)**Projekt:** Ja

Überlegen Sie sich, basierend auf Ihren Erfahrungen aus dem Projekt, neue Funktionalität für die *Movie Manager* Anwendung, die Ihre KommilitonInnen im nächsten Jahr im Rahmen eines Projekts umsetzen können. Beschreiben Sie die neue Funktionalität in textueller Form. Gehen Sie bei Ihrer Beschreibung insbesondere ein auf

- die dafür benötigten Systemfunktionen
- die zum Umsetzen der Systemfunktionen notwendigen Daten
- die Änderungen an der Benutzungsschnittstelle, und dadurch neu zu erstellende Arbeitsbereiche bzw. Änderungen an bestehenden Arbeitsbereichen.

Ergebnis:

Speichern Sie bitte Ihre Ergebnisse als PDF bis **Montag 18.01.2015 um 10.00 Uhr** in Moodle