

## Übungsblatt 11 (22.12.2015)

### Qualitätsanforderungen, Komponententest, Implementierung, Code Coverage, System- und Abnahmetest, Abschlusspräsentation

In dieser Übung:

- ✓ Spezifizieren Sie Qualitätsanforderungen an den *Movie Manager* Webclient.
- ✓ Spezifizieren Sie Komponententestfälle für den *Movie Manager* Webclient.
- ✓ Implementieren Sie Ihren erstellten Entwurf für den *Movie Manager* Webclient.
- ✓ Implementieren Sie Komponententests als JUnit Testfälle.
- ✓ Protokollieren Sie die Ausführung Ihrer Komponententests.
- ✓ Überprüfen Sie die Codeüberdeckung Ihrer Komponententestfälle.
- ✓ Führen Sie Systemtests aus und einen Abnahmetest durch.
- ✓ Bereiten Sie die Abschlusspräsentation vor.

#### Hinweise zur Bearbeitung und Abgabe von Übungsblatt 11:

Die Bearbeitung des 11. Übungsblatts erfolgt über einen längeren Zeitraum. Bitte beachten Sie, dass Sie Ihre Lösung des **11. Übungsblattes bis Montag, den 11.01.2016** in Moodle abgeben.

#### Abschlusspräsentation der Webausleihe

Sie präsentieren Ihre Webausleihe (*Movie Manager* Anwendung und *Movie Manager* Webclient) Ihrem Auftraggeber, der Firma *MovieExperience*, um diesen davon zu überzeugen, Ihr Produkt an die KundInnen auszuliefern. Da für eine Demo leider keine Zeit ist, präsentieren Screenshots der Abnahmetests (siehe unten).

Die Abschlusspräsentationen aller Teams finden während der Vorlesung und Zentralübung am **Dienstag, dem 12.01.2016** statt. Bitte stellen Sie sicher, dass alle Mitglieder Ihres Teams zu diesen Terminen anwesend sind.

#### Aufgabe 11.1: Qualitätsanforderungen an den Webclient

Präsenz: Ja	Punkte: 3	Team: Ja(4)	Projekt: Ja	Testat
-------------	-----------	-------------	-------------	--------

Erstellen Sie **mindestens drei Usability-Anforderungen und mindestens drei weitere Anforderungen zu unterschiedlichen Qualitätsattributen** für den *Movie Manager* Webclient. Legen Sie dafür Elemente vom Typ „NonFunctionalRequirement“ im „ReqSpec Document“ in dem Abschnitt „NonFunctionalRequirements“ in UNICASE an. Verlinken Sie ggf. die Qualitätsanforderungen mit anderen Anforderungselementen (z.B. User Task). Ordnen Sie die Anforderungen in der Beschreibung (Textfeld „Description“) einer der 4 Ebenen zu: Aufgaben-, Domänen-, Interaktions- und

Systemebene. Beachten Sie, dass die Qualitätsanforderungen messbar formuliert sind, d.h. dass klar ist, wie man sie überprüft. Begründen Sie, wie gut Ihre *Movie Manager* Webclient Implementierung die erstellten Qualitätsanforderungen erfüllt. Nutzen Sie für Ihre Begründungen Elemente vom Typ Rationale in Ihrem UNICASE Projekt.

### Ergebnis:

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

Bitte speichern Sie eine .zip-Datei bis **Montag 11.01.2016 um 10.00 Uhr** in Moodle bestehend aus:

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

#### Hinweise zur Aufgabe 11.2 bis 11.5

Zur Sicherstellung der Qualität Ihrer Lösung gehen Sie bei der Implementierung des *Movie Manager* Webclients nach einem Test-First ähnlichen Prinzip vor, d.h.:

1. Sie spezifizieren Komponententests für die Funktionalität des *Movie Manager* Webclients. (Aufgabe 11.2)
2. Auf Basis Ihres Entwurfs implementieren Sie den *Movie Manager* Webclient. (Aufgabe 11.3)
3. Sie führen die spezifizierten Komponententests aus, verbessern ggf. Ihre Implementierung, prüfen die Test-Code-Überdeckung und verbessern ggf. Ihre Tests sowie die Implementierung. (Aufgabe 11.4)
4. Sie führen Systemtests und einen Abnahmetest für Ihren Webclient aus bzw. durch. (Aufgabe 11.5)

Die folgenden Aufgaben 11.2 bis 11.5 hängen daher stark zusammen. Geben Sie nach der Bearbeitung der Aufgaben Ihre Lösungen **nur bei Aufgabe 11.5** in Moodle ab.

### Aufgabe 11.2: Spezifikation von Komponententestfällen für den Webclient

Präsenz: Ja	Punkte: 4	Team: Ja(4)	Projekt: Ja	Testat
-------------	-----------	-------------	-------------	--------

Im Folgenden werden die Komponententestfälle für alle Operationen *Loan* und *Return* spezifiziert. Dazu wird zustandsbasierter Test verwendet, um die Zusammenhänge zwischen den Attributen *loanable* und *loaned* und den Operationen darzustellen.

1. Beschreiben Sie zuerst das Zustandsdiagramm für die Attribute *loanable*, *loaned* der Klasse *Movie* und die Operationen *Loan/ReturnMovie*. Es muss berücksichtigt werden, dass verliehene Filme ausleihbar bleiben müssen. In einem Zustand unerlaubte Operationsaufrufe führen in den gleichen Zustand zurück. Kennzeichnen Sie diese mit einem „u“.
2. Bei der Klasse *Series* (und analog *Season*) ist zu berücksichtigen, dass die Operationen *Loan* bzw. *Return* alle Teile (*Season* oder *Episode*) auch auf verliehen bzw. zurückgegeben setzen. Beschreiben Sie das Zustandsdiagramm für die Klasse *Series* und die Operationen *Loan/ReturnSeries*. Führen Sie dazu noch zwei Attribute *allpartsloaned: bool* und *partsloaned: bool* ein. Das Attribut *allpartsloaned* ist genau dann wahr, wenn alle Staffeln und alle Episoden der Serie verliehen sind. Das Attribut *partsloaned* ist genau dann wahr, wenn mindestens eine

Staffel oder mindestens eine Episode der Serie verliehen ist. Berücksichtigen Sie die beiden Attribute neben `loaned` und `loanable` in den Zuständen.

3. Leiten Sie nun aus dem Zustandsdiagramm für *Series* 1-3 Operationsfolgen ab, die die wesentlichen Zustände, wenn `Loan` bzw. `Return` erlaubt bzw. unerlaubt sind, abdecken.
4. Spezifizieren Sie diese Operationsfolgen als logische Komponententestfälle in UNICASE als Element vom Typ „TestCase“ in dem Abschnitt „Component Tests“ im „Test Document“. Beschreiben Sie für jeden Testfall die benötigte „Infrastructure“, d.h. Stubs und Treiber, die Sie für die Testfälle benötigen, sowie jeweils die Pre- und Postcondition für den Testfall. Beachten Sie, dass Sie zur Durchführung der Änderungen auf Teilen einer Serie Stubs für diese Teile brauchen. Achten Sie dabei darauf, dass Sie sowohl in den einzelnen TestSteps als auch bei den TestCases jeweils einen aussagekräftigen Beschreibungstext formulieren, der deutlich macht, was der Testfall testet. Beschreiben Sie für jeden Testschritt die Vor- und Nachbedingung unter „Description“ und Ausnahmen unter „Exception“. Orientieren Sie sich bei Ihrer Lösung an den vorgegebenen Testfällen der Klasse *Movie*.

### Ergebnis:

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

## Aufgabe 11.3: Implementierung des MovieManager Webclients

Präsenz: Ja	Punkte: 15	Team: Ja(4)	Projekt: Ja	Testat
-------------	------------	-------------	-------------	--------

Ziel dieser Aufgabe ist die Implementierung des *Movie Manager* Webclients. Nutzen Sie für die Implementierung Ihr Wissen aus den Aufgaben 3.1 und 4.1.

Implementieren Sie den von Ihnen auf Übungsblatt 6 erstellten GUI-Entwurf und das auf Übungsblatt 10 erstellte Dialogmodell und Entwurfsklassendiagramm. Wichtig ist, dass Ihre Entwürfe konsistent mit der Implementierung sind. Sollten Sie während der Implementierung von Ihren ursprünglichen Entwürfen abweichen, so dokumentieren Sie diese Abweichung als Rationale in UNICASE mit Hilfe eines Issues und passen Sie Ihre Entwürfe entsprechend an.

### Ergebnis:

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

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

## Aufgabe 11.4: Implementierung und Ausführen der Komponententest, JUnit, Code Coverage für MovieManager Webclient

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

In dieser Aufgabe implementieren Sie Komponententests und überprüfen und dokumentieren Sie die Codeüberdeckung Ihrer implementierten Komponententests. Gehen Sie dazu wie folgt vor:

1. Testen Sie den von Ihnen erstellten Code aus Aufgabe 11.2. Implementieren Sie dafür die Testfälle, die Sie in Aufgabe 11.1 spezifiziert haben, als JUnit Tests. Beachten Sie für die Implementierung die Hinweise zum Einrichten von JUnit im Tutorial **10-UnitTest-Webclient.pdf**. Passen Sie gegebenenfalls die in UNICASE spezifizierten Testfälle entsprechend der von Ihnen durchgeführten JUnit Test Implementierung an. Dokumentieren Sie die Ergebnisse Ihrer Testdurchführung mit Testprotokollen in UNICASE. Korrigieren Sie gefundene Fehler, sodass alle Testfälle erfolgreich durchgeführt werden können.

2. Überprüfen Sie für die von Ihnen in Aufgabe 11.3.1 implementierten Komponententests die Codeüberdeckung mit Hilfe des Eclipse Plug-Ins EclEmma. Dokumentieren Sie die ermittelte Überdeckung der getesteten Operationen in einer Textdatei.
3. Fügen Sie weitere Testfälle hinzu, sodass eine 100% Abdeckung der getesteten Operationen erzielt wird. Dokumentieren Sie durch wiederholtes Ausführen der Codeüberdeckungsprüfung die resultierende Abdeckung Ihres Codes in einer Textdatei. Falls Sie keine 100% Abdeckung erzielen, begründen Sie dies in der Textdatei.
4. Aktualisieren Sie die spezifizierten Komponententestfälle in Ihrem UNICASE Projekt mit den Änderungen aus Aufgabe 11.3.3.

**Ergebnis:**

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

Bitte speichern Sie Ihre Änderungen am Quellcode auf dem Git Server und die Codeüberdeckung vor und nach dem Anpassen und Begründung als .pdf

### Aufgabe 11.5: Durchführen und Protokollieren von Systemtests für MovieManager Webclient und Abnahmetest

Präsenz: Nein	Punkte: 8	Team: Ja(4)	Projekt: Ja	Testat
---------------	-----------	-------------	-------------	--------

In dieser Aufgabe führen Sie Systemtests für den *Movie Manager* Webclient und einen Abnahmetest durch. Gehen Sie wie folgt vor:

1. Testen Sie auf Basis der Systemtests aus Aufgabe 7.2 und des aktuellen GUI Ihren *Movie Manager* Webclient. Korrigieren Sie alle gefundenen Fehler. Erstellen Sie für die 3 wichtigsten Testfälle jeweils ein Testprotokoll in UNICASE. Dokumentieren Sie für diese Testfälle das Ergebnis Ihrer Testdurchführung in den Testprotokollen. . Begründen Sie als Rationale in UNICASE, warum Sie diese 3 Testfälle ausgewählt haben.
2. Führen Sie für Ihren *Movie Manager* Webclient einen Abnahmetest mit den folgenden von Ihrem Auftraggeber *MovieExperience* vorgegeben Testfällen durch. Korrigieren Sie alle dabei gefundenen Fehler.:
  1. Legen Sie in der *Movie Manager* Anwendung eine Serie mit mindestens zwei Staffeln an. Legen Sie für jede Staffel mindestens zwei Episoden an. Stellen Sie die Serie als verleihbar ein.
  2. Führen Sie in Ihrem *Movie Manager* Webclient die folgenden Schritte durch. Fertigen Sie von jedem der untenstehenden Testschritte a bis e mindestens einen Screenshot des *Movie Manager* Webclients an. Der Screenshot sollte die Ausgangssituation, die Endsituation und den Ablauf der NutzerInnen deutlich machen.
    - a) Leihen Sie Serie aus.
    - b) Geben Sie die Serie zurück.
    - c) Leihen Sie nur einen Teil der Serie aus, z.B. eine der Staffeln.
    - d) Versuchen Sie die Serie zu leihen.
    - e) Geben Sie den geliehenen Teil der Serie zurück.

**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 Ihre **Ergebnisse aus Aufgabe 11.1, 11.2, 11.3 und 11.4** in einer .zip-Datei bis **Montag 11.01.2016 um 10.00 Uhr** in Moodle bestehend aus:

- 6 exportierten Projekten „movies“, „movies.edit“, „movies.ui“, „movies.web“, „movies.emfstore“ und „movies.web.model“
- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)
- Codeüberdeckung vor und nach dem Anpassen und Begründung als .pdf

### Aufgabe 11.6: Vorbereitung – Abschlusspräsentation

Präsenz: Nein	Punkte: 6	Team: Ja(2 x 4)	Projekt: Ja	Testat
---------------	-----------	-----------------	-------------	--------

In den Abschlusspräsentationen präsentieren jeweils zwei 4er Teams gemeinsam die Ergebnisse Ihres Entwicklungsprojekts. Ihr 4er Team präsentiert mit dem 4er Team, das auch die Inspektion Ihres UNICASE Projekts durchgeführt hat. In der Abschlusspräsentation vergleichen und bewerten Sie die Ergebnisse des Projekts. Gehen Sie wie folgt vor:

1. Erarbeiten Sie Unterschiede und Gemeinsamkeiten aus den erstellten Entwurfsartefakten beider Teams, insbesondere aus der UI-Struktur des *Movie Manager* Webclients und dem Entwurfsklassendiagramm des *Movie Manager* Webclients.
2. Erstellen Sie eine Präsentation **von 15 Minuten Dauer**, in der
  - jedes 4er Team die Screenshots für die Tests aus Aufgabe 11.5.2 zeigt. Erklären Sie dabei die wesentlichen Unterschiede der beiden Lösungen in Bezug auf die Anforderungen und bei der für die NutzerInnen sichtbaren Umsetzung.
  - Sie die Unterschiede und Gemeinsamkeiten des Entwurfs vorstellen. In der Präsentation sollen die wesentlichen Aspekte Ihres Entwurfs und Ihrer Implementierung im Vergleich zum anderen Team dargestellt werden. Verwenden Sie dazu **nur die relevanten Ausschnitte** der Diagramme (Entwurfsklassendiagramm und UI-Struktur), d.h. machen Sie die Unterschiede und Gemeinsamkeiten in den Diagrammen lesbar auf den Folien deutlich.
  - Sie sich für drei Qualitätsattribute der beiden Teams (vgl. Aufgabe 11.1) entscheiden und erklären bzw. begründen, welche Lösung für welches Qualitätsattribut besser geeignet ist.
  - Sie Ihre Erfahrungen (zusammengestellt aus beiden Teams) aus dem Projekt zur Entwicklung der Webausleihe darstellen. Beantworten Sie dazu die folgenden Fragen:
    - Welche Beschreibungstechnik fanden Sie am hilfreichsten?
    - Welche Beschreibungstechnik fanden Sie am wenigsten hilfreich?
    - Welcher Teil (Movie Manager Anwendung oder Webclient) war einfacher zu entwickeln?
    - Was würden Sie beim nächsten Projekt mit der der Firma *MovieExperience* anders machen?

Achten Sie auf eine gleichmäßige Aufteilung der Präsentationszeit auf beide Teams. Bereiten Sie sich sorgfältig auf die Präsentation vor und sprechen Sie sich mit dem anderen 4er Team ab, wann welches Team was präsentiert, um einen reibungslosen Ablauf Ihrer Präsentation zu gewährleisten.

### Ergebnis:

Speichern Sie bitte die Folien Ihrer Präsentation als PDF-Datei bis **Montag 11.01.2016 um 10.00 Uhr** in Moodle.



**Wir wünschen Ihnen und Ihrer Familie frohe Festtage und  
für das neue Jahr Gesundheit, Glück und Erfolg!**

