Qualitätssicherung (QS) VU: Beispiel „Review“

Leitfaden

Im Rahmen des Übungsteils der VU Qualitätssicherung beschreibt dieses Dokument die von den Teilnehmern durchführenden Aufgaben; die Ergebnisse der Übungsaufgaben sind in diesem Dokument aufzuzeichnen.

Aufgabenstellung

Das Aufgabenpaket für den Aufgabenbereich „Reviews“ besteht aus den Dokumenten „Aufgabenstellung_Restaurant“ (herunterzuladen im TUWEL) und der Abgabedatei (diese Datei). Die Ergebnisse der Aufgaben sind direkt hier elektronisch auszufüllen und als pdf im TUWEL abzugeben.

- Stellen Sie aus dem Leitfaden Ihre persönliche Kopie her; Sie können die Aufgaben mit Kollegen diskutieren; die Aufgaben sind jedoch von jedem Studenten **individuell auszuarbeiten** und abzugeben.

  Name: **Manuel Mausz**
  MatrikelNr: **0728348**

- Lösen Sie die in diesem Dokument enthaltenen Aufgaben (Review-Ergebnisse in Tabellen eintragen).
- Füllen Sie dieses Dokument elektronisch aus (handschriftliche Abgaben werden nicht angenommen).
- Geben Sie im TUWeL **rechtzeitig** ab
- Pro Person ist eine PDF Datei abzugeben.
- Bei Fragen zum Beispiel:
  - Posten Sie zuerst im TUWeL Forum.
  - Dann wenden Sie sich bitte an einen Tutor.
  - Schreiben Sie an unsere Service-Mailadresse: qs_admin@qse.ifs.tuwien.ac.at
  - Bei allfälligen **schwerwiegenden** Problemen wenden Sie sich an die Übungsleitung.

In dieser Abgabe sind zu den Anforderungen und Entwicklungsmodellen Reviews durchzuführen. Ergebnisse dieser Reviews sind Listen mit „Issues“, das sind relevante Abweichungen von einem korrekten Dokument; für die Aufzeichnung der „Issues“ stehen in diesem Dokument leere Tabellen zur Verfügung.

2a. Software Requirements Review der Anforderungen (3 Punkte)

Vorbedingung für Qualitätssicherung (etwa systematische Tests) sind verständliche und testbare Anforderungen. In dieser Übungsaufgabe sollen Sie den Typ von konkreten Anforderungen überprüfen, außerdem sollen Kommentare bezüglich Verständlichkeit, Mess/Testbarkeit und Rückverfolgbarkeit aufgezeichnet werden.

1. Ermitteln Sie den **Typ** von allen Anforderungen der Aufgabenstellung, überprüfen Sie anschließend 6 dieser Anforderungen auf folgende Qualitätskriterien, **Sie müssen alle Typen abdecken**.

   - **Typ**
     - *Functional* – Features, Fähigkeiten, Sicherheitsvorrichtungen
     - *Usability* – Menschliche Faktoren, Hilfe, Dokumentation
     - *Reliability* – Häufigkeit eines Ausfalls, Vorhersehbarkeit, Wiederherstellbarkeit
     - *Performance* – Antwortzeiten, Durchsatz, Genauigkeit
     - *Supportability* – Anpassungsfähigkeiten, Wartung, Konfigurationen

   - Rückverfolgbarkeit
     - Wird die Anforderung in der Projektbeschreibung verlangt? Liegt sie im relevanten Bereich des Systems?
     - Anforderungen sollen keine technischen Entwurfs- oder Implementationsvorschläge enthalten!
     - Ist das Verhalten für alle Normalfälle und alle relevanten Fehlerfälle in der Projektbeschreibung spezifiziert?
     - Mögliche Ergebnisse:
       - Nein, die Anforderung ist nicht in der Projektbeschreibung spezifiziert
       - Die Anforderung technisch viel zu ausgeführt!
       - Rückverfolgbar aber kein Fehlerfall spezifiziert.

   - Überprüfbarkeit (aus Sicht des Auftraggebers)
     - Ist die Anforderung durch z.B. einen Test verifizierbar?
     - Kann der Test bereits vor der Implementierung realisiert werden?
     - Ist das Ergebnis des Tests leicht festzustellen? Messbar? Auch ohne Software?
     - Manuell oder auch automatisiert testbar?
     - Wenn ja, dann wie?
     - Wenn nein, warum nicht? Änderungsvorschläge festhalten.
     - Mögliche Ergebnisse:
       - Ja, aber in der steht „X“: Widerspruch!
       - Ja, durch einen Test realisierbar: vor dem Test muss die Datenbank in den Zustand X gebracht werden.
       - Nein, „...leicht änderbar.“ ist nicht testbar. Anforderung wäre besser Überprüfbar durch ändern auf „...durch Konfigurationsdatei zur Laufzeit änderbar.“
       - Ja, jedoch nicht automatisierbar, test muss manuell an der laufenden Applikation durchgeführt werden.
       - Nein, mit Umformulierung: „Das System muss X in <=2 Sekunden ausführen“

Überprüfen Sie jede Anforderung und begründen Sie ihre Antwort in eigenen Worten! Falls eine Anforderung Ihrer Meinung nach nicht testbar oder messbar ist, schlagen Sie vor wie dies erreicht werden könnte.

Nr.	Typ	Rückverfolgbarkeit	Überprüfbarkeit
27	Supportability	Ja, zur Projektbeschreibung rückverfolgbar: „Ein kompliziertes Rezept kann aus einfacheren Rezepten bestehen “ im dritten Absatz.	Ja, durch automatisierte Tests verifizierbar. Datenbanktest: Integritätsverletzung muss erzeugt werden für Fehlerfall. Mit Umformulierung noch besser Überprüfbar: „...“
12	Functional (possible Use Case)	Ja, aber in der Beschreibung steht „ *Zu einer Bestellung über **€250.-** ist eine Anzahlung von 10% zu leisten.*“: Widerspruch!	Ja. Die Anforderung ist zudem auch automatisiert Testbar.
15	Usability	Nein, die Anforderung ist nicht in der Projektbeschreibung spezifiziert.	Ja, jedoch nicht automatisierbar, Test muss manuell an der laufenden Applikation durchgeführt werden. Zudem hängt das Testergebnis von der Sehstärke des Testers ab.
3	Reliability	Nein, die Anforderung ist nicht in der Projektbeschreibung spezifiziert.	Ja, das Speichern der Variablen in der Datenbank ist testbar. Ob die Werte zwangsläufig Sinn ergeben, ist nicht test- und nur sehr schwer überprüfbar.
17	Performance	Nein, die Anforderung ist abgesehen von der Spezifikation der Systemintegrität nicht in der Projektbeschreibung spezifiziert.	Ja, die Anforderung ist auch automatisiert Testbar. Vermutlich muss zudem für einige Tests die Datenbank entsprechend mit Daten vorbereitet werden.
16	Supportability	Nein, die Anforderung ist nicht in der Projektbeschreibung spezifiziert.	Ja, ist höchstwahrscheinlich automatisiert Testbar. Hierfür gibt es üblicherweise Emulatoren für Handhelds.
2	Reliability	Ja, aber in der Beschreibung steht „*Zu jeder Rechnung für einen Kunden werden die Endsumme und der verantwortliche **Koch** vermerkt*“: Widerspruch!	Ja, das Berechnen und Speichern der Variablen in der Datenbank ist testbar.

2b. Design Review der Anwendungsfälle (4 Punkte)

Jede Funktionale Anforderung wird in ein oder mehrere Anwendungsfälle übergeführt. Änderungen in den Anforderungen ergeben Änderungen in den Anwendungsfällen, diese sollen jetzt einem Review unterzogen werden.

1. Gehen Sie die Anforderungen systematisch durch und markieren Sie 3 verschiedene Arten von Wörter:
    - **Akteure**, z.B. Kunde
    - Entitäten, z.B. Speise
    - **Operationen**, z.B. aktualisiert

- Die vorhandenen Anwendungsfälle wurden durch die **Akteure** und deren **Operationen** welche in den Anforderungen und der Projektbeschreibung vorhanden sind, gebildet.

2. Überprüfen Sie jede vorhandene **Anwendungsfallbeschreibung (nur Beschreibung, ohne Diagramm)** auf folgende Qualitätskriterien:
    - Ist jeder Anwendungsfall einer Anforderung zugeordnet?
    - Ist jeder Anwendungsfall in der Projektbeschreibung zu finden?
    - Ist jeder Anwendungsfall notwendig, plausibel, korrekt und vollständig?
    - Ist jeder Anwendungsfall mit dem richtigen Akteur verbunden?
    - Sind die Assoziationen zu anderen Anwendungsfällen sinnvoll?
    - Mögliche Ergebnisse:
        - Anwendungsfall hat keine zugehörige Anforderung.
        - Anwendungsfall hat zugehörige Anforderung: 23
        - Fehler in der Notation des Anwendungsfalles (Assoziation fehlt, ist inkorrekt oder überflüssig).
        - Anwendungsfall ist unvollständig (z.B. Akteur fehlt, weiterer Anwendungsfall ist notwendig).
        - Fehler in Schritt 3 der Szenario Beschreibung: Anwendungsfall X existiert nicht
        - Fehler in den Vorbedingungen: X ist keine gültige Vorbedingung!
        - ...

Name	Review Issues der Anwendungsfallbeschreibungen
Produkt verkaufen	- Fehler in Notation: “Beispiele anzeigen” includes “Auswahl treffen” ist falsch. - “Beispiele mit Lösungen anzeigen” extends “Beispiele anzeigen” Assoziation fehlt. - Anwendungsfall wird auch von Anforderung 28 verwendet. - Eine weitere Generalisierung „Anwendungsfall 3“ wäre möglich - Schlechter Name! Anwendungsfälle sollten mit einem starken Verb anfangen! - Nicht zu den Anforderungen oder Beschreibung Rückverfolgbar!
verkaufe Speisen	- Anwendungsfall ist unvollständig: Bei Schritt 3 fehlt Extension Point zu „leiste Anzahlung“ - Anwendungsfall ist unvollständig: Die Angabe wann das Essen stattfindet und ob es im Restaurant gegessen oder mitgenommen wird fehlt. - Nicht zu den Anforderungen oder Beschreibung Rückverfolgbar! - Anwendungsfall ist fehlerhaft: Sekundärer Akteur „Einkäufer“ kommt nicht vor. - Anwendungsfall ist unvollständig: Sekundärer Akteur sollte „Kunde“ und „Koch“ sein.
drucke Einkaufsliste	- Anwendungsfall hat zugehörige Anforderung: #8 - Nachbedingung ist fehlerhaft: Es wird die Einkaufsliste gedruckt - Anwendungsfall ist unvollständig: Laut Beschreibung berücksichtigt das System auch den Lagerstatus
erstelle Tagesbilanz	- Anwendungsfall ist fehlerhaft: Laut Beschreibung und Anforderung wird eine Wochenbilanz erstellt. Unter Annahme „erstelle Wochenbilanz“: - Anwendungsfall hat zugehörige Anforderung: #5 - Anwendungsfall ist unvollständig: Die Nachbedingung fehlt: „Integrationscheck durchgeführt“ - Anwendungsfall ist unvollständig: Bei Schritt 5 fehlt Extension Point zu „erstelle Korrekturbuchung“ - Anwendungsfall ist fehlerhaft: Schritt 3: Der „Buchhalter“ muss die Schritte wiederholen.

3. **Review der Anwendungsfallbeschreibungen mit dem dazugehörigen Diagram**. Überprüfen Sie die vorhandenen Anwendungsfallbeschreibungen mit dem Diagramm auf Übereinstimmung, Sie können auch das Diagramm mit der Projektbeschreibung vergleichen: gibt es Unstimmigkeiten? Liste Sie etwaige Fehler bzw. Issues hier auf:

- **vergleiche Beträge**: hat anderen Namen in der Beschreibung: „Beträge vergleichen“
- **do X**: Szenario weißt auf einen Extension Point hin welcher im Diagramm nicht existiert!

- verkaufe Speisen: sekundärer Akteur fehlt.
- verkaufe Speisen: „leiste Anzahlung“ ist nicht in Anwendungsfallbeschreibung vorhanden.
- Condition von leiste Anzahlung: Laut Anforderungen muss die Gesamtsumme 500€, laut Beschreibung 250€ überschreiten
- drucke Einkaufsliste: Laut Szenario ist „Vorbestellung auswerten“ kein Anwendungsfall.
- drucke Einkaufsliste: Laut Szenario ist Akteur „Koch“ an Anwendungsfall beteiligt.
- erstelle Tagesbilanz: Extension Point muss „drucke fehlende Rechnungen“ heißen.

- Generalisierung der Akteure ist in der Beschreibung nicht spezifiziert.

2c. Design Review des Domänenmodells (4 Punkte)

1. Gehen Sie die Projektbeschreibung systematisch durch und markieren Sie 3 verschiedene Arten von Wörter:
   - Akteure, z.B. Kunde
   - **Entitäten**, z.B. Speise
   - **Operationen**, z.B. aktualisiert

- Die vorhandenen Klassen im Domänenmodell wurden durch die **Entities** und deren **Operationen** welche in den Anforderungen und der Projektbeschreibung vorhanden sind, gebildet.

2. Überprüfen Sie **8 Klassen** des Domänenmodells auf folgende Qualitätskriterien:
   - Ist jede Klasse einer Anforderung zugeordnet?
   - Ist jede Klasse in der Projektbeschreibung zu finden?
   - Haben die Assoziationen korrekte Multiplizitäten?
   - Ist jede Klasse notwendig, plausibel, korrekt und vollständig?
   - Sind die Assoziationen zu anderen Klassen sinnvoll?
   - Mögliche Ergebnisse:
     - Klasse hat keine zugehörige Anforderung.
     - Klasse hat zugehörige Anforderung: 23
     - Fehler in der Notation des Klasse (Assoziation fehlt, ist inkorrekt oder überflüssig).
     - Attribut ist falsch modelliert! (weitere Klassen sind notwendig).
     - Klasse ist unvollständig: AttributX und AttributY fehlt.
     - Fehler! Anforderung besagt es kann keine Speise ohne Rezept geben
     - ...

- Falls Sie einen Fehler finden tragen Sie diesen in der Issues Spalte ein. In dieser Review können nicht nur Fehler, sondern auch Fehlende Klassen gefunden werden, markieren Sie diese mit „NEW“.

Klassenname	Issues (Kandidaten für Diskussion)
Klasse1 Assoziationen: Creates: Klasse1 * –> 1 Klasse2	- Klasse ist nicht vollständig: Attribut „payed“ fehlt Assoziation „Creates“ hat falsche Multiplizität, richtig: Creates: Klasse1 1..* –> 1 Klasse2 Sollte gerichtet sein! - Klasse hat zugehörige Anforderung, sollte jedoch laut Projektbeschreibung mehr Assoziationen haben! o Belongs to: Klasse1 1 <– 1..* Klasse3
Zutat	- Klasseattribute „verdorben“ fehlt. - Asszіationsklasse zu Assoziation Zutat --- Einkauf: menge
Lager	- Klasse ist nicht vollständig. Assoziation fehlt: enthält: Lager 0..1 --- * Zutat Asszіationsklasse zu neuer Assoziation „enthält“: lagerMenge
NEW Tagesplan NEW Assoziationen: erhält: Koch 1 --- 1 Tagesplan enthält: Bestellung 1..* --- 0..1 Tagesplan	- Klasse laut Anforderung aus Beschreibung.
Rezept	- Klassenattribut „zutatenMenge“ ist fehlerhaft: Laut Beschreibung soll jede Zutat mit zugehöriger Menge gespeichert werden. d.h. als Asszіationsklasse.

Klassenname	Issues (Kandidaten für Diskussion)
Bestellung	- Klassenattribut „take_away“ fehlt.
Sofortbestellung	- Keine zugehörige Anforderung und nicht Zurückverfolgbar.
Rechnung	- Klassenattribut „bezahlt“ fehlt.
NEW Wochenbilanz NEW Assoziationen: erzeugt: Buchhalter 1 --- 0..* Wochenbilanz	- Klasse laut Anforderung #5.

2d. Review missionskritischer Testfälle (4 Punkte)

In dieser Übungsaufgabe ist zu überprüfen, ob konkrete Testfälle verständlich und umsetzbar (durch Entwickler bzw. Tester) sind.

Überprüfen Sie **8 Testfälle** in der Liste der „Missionskritischen Testfälle“ (siehe Aufgabenstellung) mit den Anforderungen (Initiale Spezifikation, Liste der Anforderungen) auf folgende Qualitätskriterien und begründen Sie ihre Antwort in eigenen Worten.

1. Verständlichkeit / Issues
   - Sind alle Vorbedingungen notwendig und ausreichend erklärt?
   - Sind die wesentlichen Teile der Anforderungen von diesem Test ausreichend abgedeckt oder sind weitere Tests notwendig?
     - Falls die Anforderungen unzureichend abgedeckt sind, merken Sie dies mit einer kurzen Erklärung an, in diesem Beispiel sind keine neuen Testfälle zu schreiben.
   - Ist ein Fehlerfall notwendig?
   - Ist der Testfall gar kein NF sondern ein SF?
   - Passen die Aktionen im Testfall zu den angegebenen Eingabewerten?
   - Sind die Aktionen ausreichend erklärt um zu verstehen um was es in dem Testfall passiert?

2. Umsetzbarkeit
   - Ist der Testfall von einem Entwickler klar umsetzbar?
   - Falls ein Test Ihrer Meinung nach nicht vollständig umsetzbar erscheint, erklären Sie warum und schlagen Sie vor, wie dies erreicht werden könnte. (z.B. Informationen betreffend Test Umgebung)
   - Ist das erwartete Ergebnis des Tests konsistent mit den Anforderungen? Ist es entscheidbar?
   - Mögliche Ergebnisse:
     - Ja, als Unit/Integrations/Systemtest umsetzbar, jedoch nicht automatisierbar, weil ...
     - Nicht automatisierbar, User Interaktion laut Anforderung erfordert.
     - Nein, Informationen bezüglich User Interface oder anderen Architekturellen Fragen müsste feststehen.
     - Ja, Unit Test kann mit einem definiertem DB Zustand und Eingabeset durchgeführt werden und danach mittels Vergleichsdaten (Testdaten) verifiziert werden.

Test Nr.	Verständlichkeit / Issues	Umsetzbarkeit
27	-Testfall 27 und 28 decken die Anforderungen nicht vollständig ab, Anforderung 19 spezifiziert weitere Möglichkeiten: ... Fehlerfall fehlt, Vorbedingungen unvollständig, falsche Parameter, etc...	Technisch als Integrations oder Systemtest umsetzbar. Die veränderten Werte können anschließend aus der DB ausgelesen und verglichen werden. Auch Messbar durch Performance Test, jedoch aufwändig da Verteilt.
1	- In Anforderung 6 ist spezifiziert, dass nur 2 Bestellungen auf einmal bezahlt werden können. - Fehler in erwartetes Ergebnis: Die Endsumme sowie der verantwortliche Kellner wird ebenfalls zur Rechnung gespeichert (Anforderung #2) - Fehlender Fehlerfall: Rechnung ist bereits bezahlt. Rechnung kann nicht noch einmal bezahlt werden.	- Ja, automatisiert Testbar. Unittest kann mit einem definiertem DB Zustand und Eingabeset durchgeführt werden und danach mittels Vergleichsdaten (Testdaten) verifiziert werden.
3	- Vorbedingung unvollständig: Speise muss auch bestellbar sein (Anforderung #6). - Vorbedingung und Eingabeparameter unvollständig: Kunde muss existieren; Kunde.id. - Fehlender Fehlerfall: Bestellung.fürDatum darf nicht in der Vergangenheit liegen. - Fehler in erwartetes Ergebnisse: Eine Bestellung kann keinem Kellner zugeordnet werden. Nur eine Rechnung hat diese Assoziation.	- Ja, automatisiert Testbar. Unittest kann mit einem definiertem DB Zustand und Eingabeset durchgeführt werden und danach mittels Vergleichsdaten (Testdaten) verifiziert werden.
5	- Vorbedingung und Eingabeparameter unvollständig: Kunde muss existieren; Kunde.id. - Fehlender Fehlerfall: Bestellung ist bereits einer Rechnung zugewiesen. In diesem Fall ist die Rechnung schon ausgestellt und die Bestellung kann nicht mehr storniert werden.	- Ja, automatisiert Testbar. Unittest kann mit einem definiertem DB Zustand und Eingabeset durchgeführt werden und danach mittels Vergleichsdaten (Testdaten) verifiziert werden.
6	- Eingabeparameter unvollständig: Einkäufer.name - Fehler in erwartetes Ergebnis: Auch der Einkäufer wird vermerkt.	- Ja, automatisiert Testbar. Unittest kann mit einem definiertem DB Zustand und Eingabeset durchgeführt werden und danach mittels Vergleichsdaten (Testdaten) verifiziert werden.

Test Nr.	Verständlichkeit / Issues	Umsetzbarkeit
7	- Fehlender Fehlerfall: Menü existiert schon. - Vorbedingung unvollständig: Speise muss auch bestellbar sein (Anforderung #6).	- Ja, automatisiert Testbar. Unittest kann mit einem definiertem DB Zustand und Eingabeset durchgeführt werden und danach mittels Vergleichsdaten (Testdaten) verifiziert werden.
8	- Vorbedingung falsch: Es müssen keine Vorbestellungen verfügbar sein. In diesem Fall ist die Zutatenliste einfach leer. - Fehler in Parameter und Aktion: Benutzer kann mehrere zusätzliche Zutaten hinzufügen. - Fehler in Aktion: Benutzer kann Zutaten aus Vorbestellung auch modifizieren.	- Ja, als Unittest Teil umsetzbar. Jedoch muss die ausgedruckte Einkaufsliste händisch verifiziert werden.
9	- Vorbedingung falsch: „System wird per Touch-Screen mit Latex-Handschuhen bedient“ ist irrelevant. Hierfür muss eigener Testfall erzeugt werden. - Falscher Typ: Testfall ist NF. - Fehler in Parameter und Aktion: Die Menüs sind von der Speisekarte unabhängig und auch nicht zu dieser vermerkt.	- Ja, automatisiert Testbar. Unittest kann mit einem definiertem DB Zustand und Eingabeset durchgeführt werden und danach mittels Vergleichsdaten (Testdaten) verifiziert werden.
10	- Vorbedingung unvollständig: Speise muss auch bestellbar sein (Anforderung #6). - Vorbedingung und Eingabeparameter unvollständig: Kunde muss existieren; Kunde.id.	- Ja, automatisiert Testbar. Unittest kann mit einem definiertem DB Zustand und Eingabeset durchgeführt werden und danach mittels Vergleichsdaten (Testdaten) verifiziert werden.