

Use-Cases



0, Editieren

Akteure : Use-Case 1 bzw. 2

Ablauf : - Klick auf 'editieren'
 - Eingabemaske erscheint (aktuelle Daten werden abgefragt)

- Felder : Stereotyp
 Paket
 Klassenname
 Attribute (Datentypen,

Sichtbarkeit, Initialwert)

Operationen (Sichtbarkeit,
Datentypen)

- User füllt Felder aus klickt auf 'ok' in der
Eingabemaske

Nachbedingungen : Daten werden zurückgeliefert

1, Anlegen einer Klasse

Akteure : Benutzer

Ablauf : - Benutzer klickt auf 'neue Klasse'
- Use-Case 0 wird gestartet
- Falls Daten korrekt wird neues Objekt
angelegt

Nachbedingung : Neues Objekt wird angelegt

Vorbedingung : Objekt noch nicht vorhanden

2, Klasse editieren

Akteure : Benutzer

Ablauf : - Benutzer klickt auf 'editiere Klasse'
- Use-Case 0 erhält vorhandene Daten
- Benutzer ändert Daten
- Use-Case 0 gibt Daten zurück
- Falls Daten korrekt wird Objekt geändert
und gespeichert

Nachbedingung : Objekt ist geändert

Vorbedingung : Klasse vorhanden

3, Klasse löschen

Akteure: Benutzer

Ablauf: - Benutzer klickt auf „Klasse löschen“
- Sicherheitsabfrage erscheint: ‚wirklich
löschen?‘

- Wenn ‚Ja‘, Klasse löschen und alle Assoziationen, mit denen die Klasse zusammenhängt, löschen (Aufruf von Use-Case 7); sonst Abbruch des Use-Cases

Nachbedingung: Klasse ist gelöscht

4, Assoziationsdaten erfassen

Akteur: Use-Cases 5 und 6

Ablauf: Eingabemaske erscheint, zur Auswahl: Assoziation zwischen 2 oder mehr als 2 Klassen

Eingabemaske für Assoziation zwischen 2 Klassen:

- Klasse 1
 - Rolle Klasse 1
 - Multiplizität Klasse 1
 - Klasse 2
 - Rolle Klasse 2
 - Multiplizität Klasse 2
 - Assoziationstyp: zur Auswahl stehen:
 - Vererbung
 - Gerichtete Assoziation
 - Abgeleitete Assoziation
 - Qualifizierte Assoziation
 - Geordnete Assoziation
 - Verfeinerung
 - Abhängigkeit
 - Komposition
 - Aggregation
 - Mehrgliedrige Assoziation
- Beschriftung der Assoziation

Klassen: Eingabemaske für Assoziationen zwischen mehr als 2

- Klassenliste wird angezeigt

- Button ‚hinzufügen‘, um Klassen aus der Liste in die Assoziation aufzunehmen
- Für jede hinzugefügte Klasse werden Rolle und Multiplizität erfragt

Nachbedingung: Daten werden zurückgeliefert

5, Anlegen einer Assoziation

Akteur: Benutzer

Ablauf:

- Benutzer klickt auf „neue Assoziation“
- Use-case 4 wird aufgerufen
- Falls die Daten korrekt sind, wird ein neues Objekt angelegt

Vorbedingungen: Es ist noch keine Assoziation zwischen den beteiligten Klassen vorhanden

Nachbedingungen: Assoziation wurde angelegt

6, Assoziation editieren

Akteur: Benutzer

Ablauf:

- Benutzer klickt auf „Assoziation editieren“
- Use-Case 4 wird aufgerufen, dabei wird die Änderung der beteiligten Klassen deaktiviert (hierzu muss die Assoziation gelöscht werden und eine neue angelegt werden)

Nachbedingungen: Assoziation wurde mit Änderungen gespeichert

7, Assoziation löschen

Akteur: Benutzer oder Use-Case 3

Ablauf:

- Benutzer klickt auf „Assoziation löschen“
- Sicherheitsabfrage erscheint: ‚Wirklich löschen?‘
- Wenn ‚Ja‘ wird die Assoziation gelöscht, sonst Abbruch des Use-Cases

Nachbedingung: Assoziation gelöscht

8, Use-Case Daten erfassen

Akteur: Use Case 9 bzw. 10

Ablauf: Eingabemaske erscheint mit folgenden Eingabefeldern:

- Name
- Akteur
- Ablauf
- Vorbedingungen
- Nachbedingungen
- Invarianten
- Fehler
- Dialogbeispiel
- Nicht-funktionale Anwendungsanforderungen
- Variationen
- Ansprechpartner
- Regeln
- Services
- [Wunsch: Beziehungen zu anderen Use-Cases definieren]

Nach Ausfüllen der gewünschten Felder (hier müssen nicht alle Felder ausgefüllt werden, nur der Name muss eingegeben werden) schließt der Benutzer die Eingabe mit Klick auf „Okay“ ab.

Nachbedingungen: Daten wurden zurückgegeben

9, Use-Case anlegen

Akteur: Benutzer

Ablauf:

- Benutzer klickt auf „neuer Use-Case“
- Use-Case 8 wird aufgerufen
- Neuer Use-Case wird mit den übergebenen Daten angelegt und gespeichert

Nachbedingung: Use-Case ist gespeichert

10, Use-Case ändern

Akteur: Benutzer

Ablauf:

- Benutzer klickt auf „Use-Case ändern“
- Use-Case 8 wird aufgerufen, dabei werden die aktuellen Angaben zum Use-Case angezeigt
- Benutzer ändert die Angaben seinen Wünschen gemäss
- Use-Case wird mit den geänderten, von Use-Case 8 zurückgegebenen Daten gespeichert

Vorbedingung: Zu bearbeitender Use-Case existiert bereits

Nachbedingungen: geänderter Use-Case wird abgespeichert

11, Use-Case löschen

Akteur: Benutzer

Ablauf:

- Benutzer klickt „Use-Case löschen“
- Sicherheitsabfrage erscheint: ‚wirklich löschen?‘
- Bei ‚Ja‘ wird der Use-Case gelöscht, sonst Abbruch

Vorbedingung: Use-Case existiert bereits

Nachbedingung: Use-Case ist gelöscht

12, Pflichtenheftdaten editieren

Akteur: Benutzer, Use-Case 13

Ablauf: Eingabemaske erscheint, in der entweder die aktuellen Daten oder (beim Anlegen eines neuen Projekts) Default-Werte angezeigt werden:

- Zielbestimmung
- Produkteinsatz
- Produktumgebung
- Produktleistungen
- Qualitätsziele
- Testfälle / -szenarien
- Use-Cases (nur namentlich anzugeben)
- Entwicklungsumgebung
- Ergänzungen
- [Wunsch: GUI-Entwurf]

Dabei werden für alle hier aufgeführten Punkte (und deren Unterpunkte nach dem Pflichtenheft-Schema wie auf der Website zum ProPra – hier der Einfachheit halber weggelassen) Textfelder zur Eingabe aufgeführt. Ein neues Pflichtenheft wird automatisch angelegt, sobald der Benutzer ein neues Projekt anlegt, daher haben wir für diese Aktion keinen eigenen Use-Case modelliert.

Nachbedingung: geändertes Pflichtenheft ist gespeichert

13, neues Projekt anlegen

Akteur: Benutzer

Ablauf:

- Benutzer klickt auf „neues Projekt starten“
- Eventuell noch geöffnetes Projekt wird gespeichert und geschlossen (es kann also nur 1 Projekt zur gleichen Zeit geöffnet sein)
- Der Name für das neue Projekt wird erfragt
- Hat der Benutzer die Namenseingabe beendet, wird geprüft, ob es schon ein Projekt mit gleichem Namen gibt.
- Falls ja, wird der Benutzer zur Änderung aufgefordert
- Sonst: Pflichtenheft wird angelegt und Use-Case 12 wird aufgerufen

- Manager für Klassenliste, Use-Case Liste und Assoziationsliste werden angelegt

Nachbedingung: Projekt ist angelegt