

Checkliste: Use-Cases

Quelle: Heide Balzert, Lehrbuch der Objektmodellierung, 2. Auflage, Spektrum-Verlag

Ergebnisse

- **Use-Case-Diagramm**
Use-Cases, Akteure, Assoziationen, extend- und include-Beziehungen, Generalisierungsstrukturen und Systeme eintragen
- **Spezifikation der Use-Cases**
Mit Use-Case-Schablone, mittels Aktivitätsdiagrammen oder Zustandsdiagrammen

Konstruktive Schritte

Wer ist der Akteur?

- Welche Personen führen diese Aufgabe zurzeit oder künftig durch?
- Wer gibt Daten in das Softwaresystem ein? Wer erhält Ausgabedaten?
- Bedienen die Akteure das System direkt oder indirekt?
- Werden zeitliche Ereignisse modelliert? (Akteur „Zeit“ einführen)

Use-Cases kategorisieren

- Primäre, sekundäre und optionale Use-Cases unterscheiden

Standardfall eines Use-Cases spezifizieren

- Was ist der Standardfall?
- Welche Verarbeitung ist „im Normalfall“, in 80% der Fälle, durchzuführen?
- Was ist die einfachste Verarbeitung des Use-Cases?
- Welche Ergebnisse will der Akteur in den meisten Fällen erhalten?

Ablauf eines Use-Cases

- Wann beginnt der Use-Case? Welches Ereignis löst den Arbeitsablauf aus?
- Welche Eingabedaten werden benötigt?
- Welche Vorbedingungen müssen erfüllt sein?
- Welche Schritte (in welcher Reihenfolge?) sind auszuführen?
- Welche Zwischen- und Endergebnisse werden erstellt?
- Welche Nachbedingungen (Vorbedingungen anderer Use-Cases) werden sichergestellt?
- Wann endet der Use-Case?
- Wie wichtig ist diese Arbeit?

Vom Ereignis zum Use-Case

- Erstellen Sie eine Ereignisliste
- Identifizieren Sie für jedes Ereignis einen Use-Case
- Unterscheiden Sie externe und zeitliche Ereignisse

Erweiterungen, Sonderfälle

- Identifizieren Sie:
 - optionale Teile eines Use-Case
 - komplexe oder alternative Möglichkeiten
 - Aufgaben, die nur selten durchgeführt werden
- **Modellieren Sie umfangreiche Erweiterungen mit einer „extend“-Beziehung. Diese sind grundsätzlich optional.**
- Alternativ zur extend-Beziehung können diese Teile mit der Use-Case-Schablone dokumentiert werden.

Include-Beziehungen

- Komplexe Schritte als Use-Cases spezifizieren (include)
- Komplexe Use-Cases (viele Sonderfälle) in mehrere Use-Cases zerlegen und Gemeinsamkeiten mit include modellieren

Generalisierungen von Use-Cases

- Ähnliche Use-Cases als Spezialisierung eines Basis-Use-Cases modellieren
- Generalisierung nicht mit extend-Beziehung verwechseln: Spezialisierte Use-Cases sind vollständige Use-Cases, extend-Use-Cases nicht.

Analytische Schritte

„Gute“ Beschreibung

- Verständlich für den Auftraggeber
- Fachliche Beschreibung des Arbeitsablaufes
- Beschreibt Standardfall separat und Sonderfälle separat
- Maximal eine Seite

Use-Case-Namen

- Enthalten starkes Verb (spezifisch, z.B. „zeichne“, „erstelle“)
- Beschreiben Funktionalität mit wenigen Worten

Namen für Akteure

- Intuitiv verständlich und in der Terminologie des Anwendungsbereiches entnommen
- Substantiv im Singular (Administrator anstatt Administratoren)
- Keine Job-Titel von Visitenkarten, keine konkreten Namen

Akteure und Use-Cases

- Jeder Akteur soll mit mind. 1 Use-Case kommunizieren
- Ausnahme: extend- oder include-Teile

Umfang-Use-Case-Diagramm

- Maximal 10 Use-Cases

Konsistenz Use-Case-Diagramm und Schablone

- Sind alle Akteure der Schablone auch im Use-Case-Diagramm eingetragen und umgekehrt?

Fehlerquellen

- Zu kleine und damit zu viele Use-Cases
- Zu frühe Betrachtung von Sonderfällen
- Zu detaillierte Beschreibung des Use-Cases
- Verwechseln von Generalisierung und extend-Beziehung
- Use-Cases beschreiben Dialogabläufe
- Use-Cases nicht für den Auftraggeber verständlich

Use-Case-Schablone

Use-Case	Name
Ziel:	
Kategorie:	Primär: zentrale Funktionalität, kann nicht ohne funktionieren Sekundär: wichtige Funktionalität, kann aber auch ohne gehen Optional: nice-to-have
Akteure:	
Auslösendes Ereignis:	
Vorbedingung:	Was muss erfüllt sein, damit der Use case überhaupt starten kann?
Essentielle Schritte:	1. 2.
Erweiterungen:	1a 1b
Alternativen:	1a
Nachbedingung Erfolg: Nachbedingung Fehler:	(Vorbedingung für spätere Use-Cases)
Kommentare, Bem.	