



Testen mit Use Cases

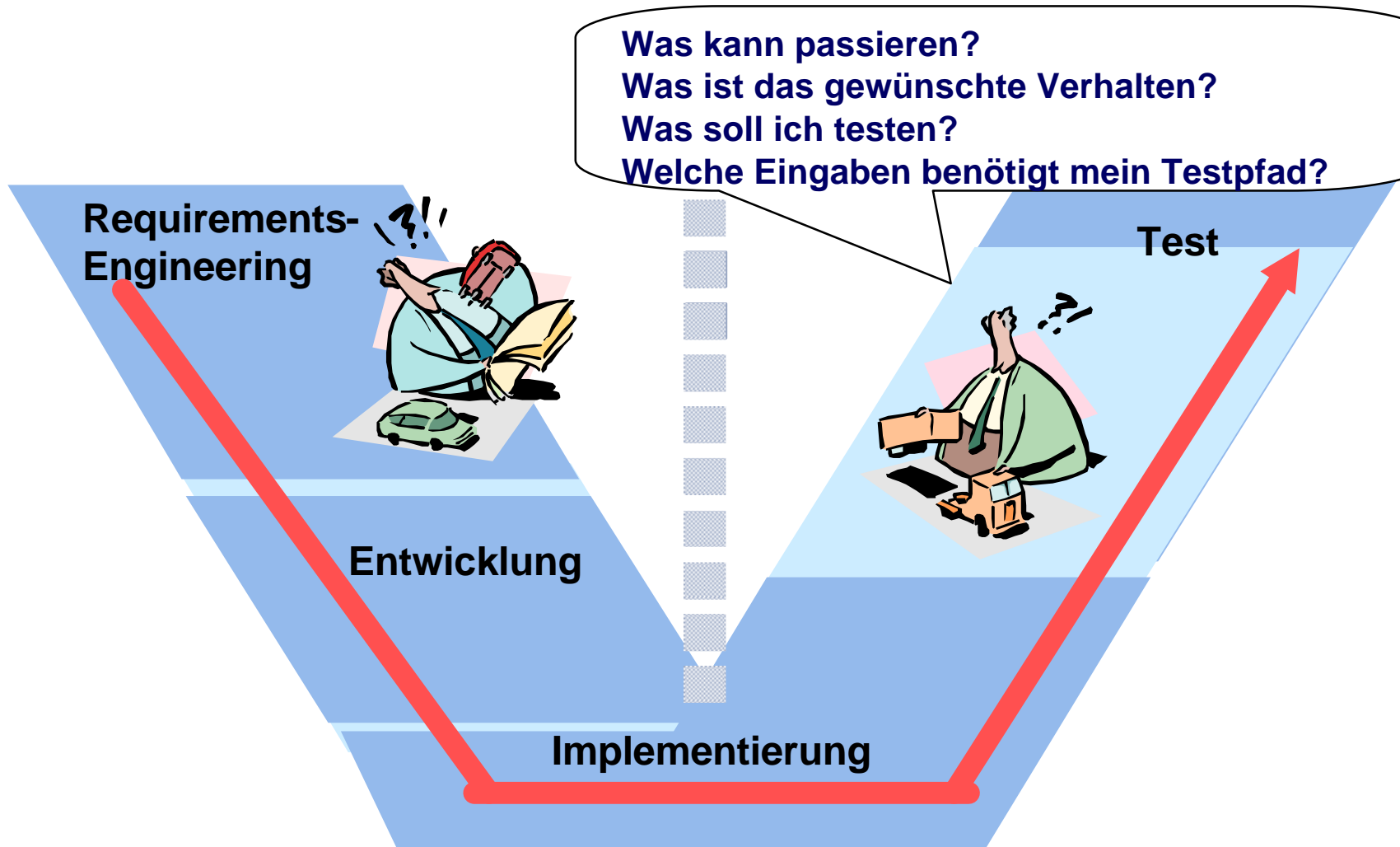
Chris Rupp

Dr. Stefan Queins

SOPHIST GROUP 

The SOPHIST GROUP logo features the text "SOPHIST GROUP" in a bold, dark blue, sans-serif font, followed by a circular sunburst icon composed of short, dark blue lines.

Das Problem



Herausforderungen für das Zusammenspiel von Requirements Engineering und Test

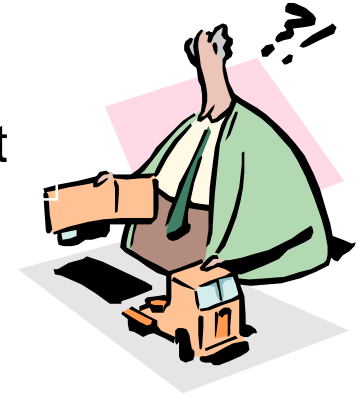


RE

> Klar definiertes funktionales Verhalten

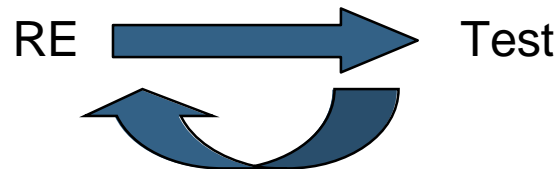


Test

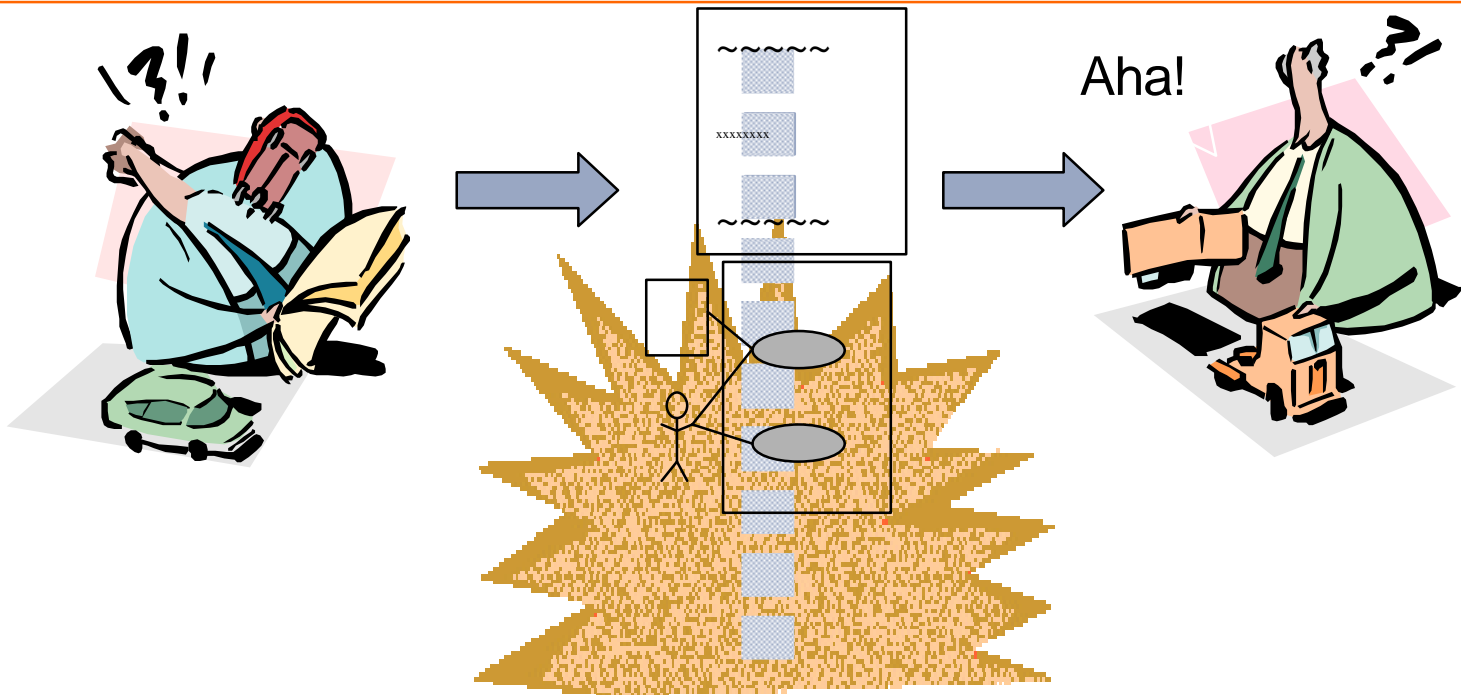


> Systematisches Ableiten von Testfällen

- Komplexität beherrschen ...
- ... auf verschiedenen Systemebenen
- vor allem: Wissen austauschen



Use Cases als Kommunikationsmittel



- Gängige Notation zur Prozessbeschreibung
- Gutes, „generelles“ Systemverständnis
- Grober Zusammenhang leicht zu verstehen (Diagramme)
- Nicht zu formell, aber trotzdem strukturiert (UML)
- Synergien mit Entwicklung / Implementierung

Use Cases -- Pros



transportieren **Wissen** -- *seit Jahrtausenden !*

fokussieren die **Benutzersicht** *im Gegensatz zu „inside-out“*

beschreiben **End-to-End Abläufe** -- *nicht interne
Teilschritte*

setzen Details in **Kontext** -- *ein Mittel gegen
Mehrdeutigkeit*

Use Cases -- zu beachten



benötigen eine **Kontextabgrenzung** *zu Nachbarsystemen*

im Prozess

viele Anwendungsmöglichkeiten

*Erhebung, Analyse,
Spezifikation,
Design, Test, ...*

es gibt

viele Varianten

*„sketchy“ bis analytisch
Use-, Misuse-, Failure-Cases
black-box / white box*

beschreiben

Abhängigkeiten

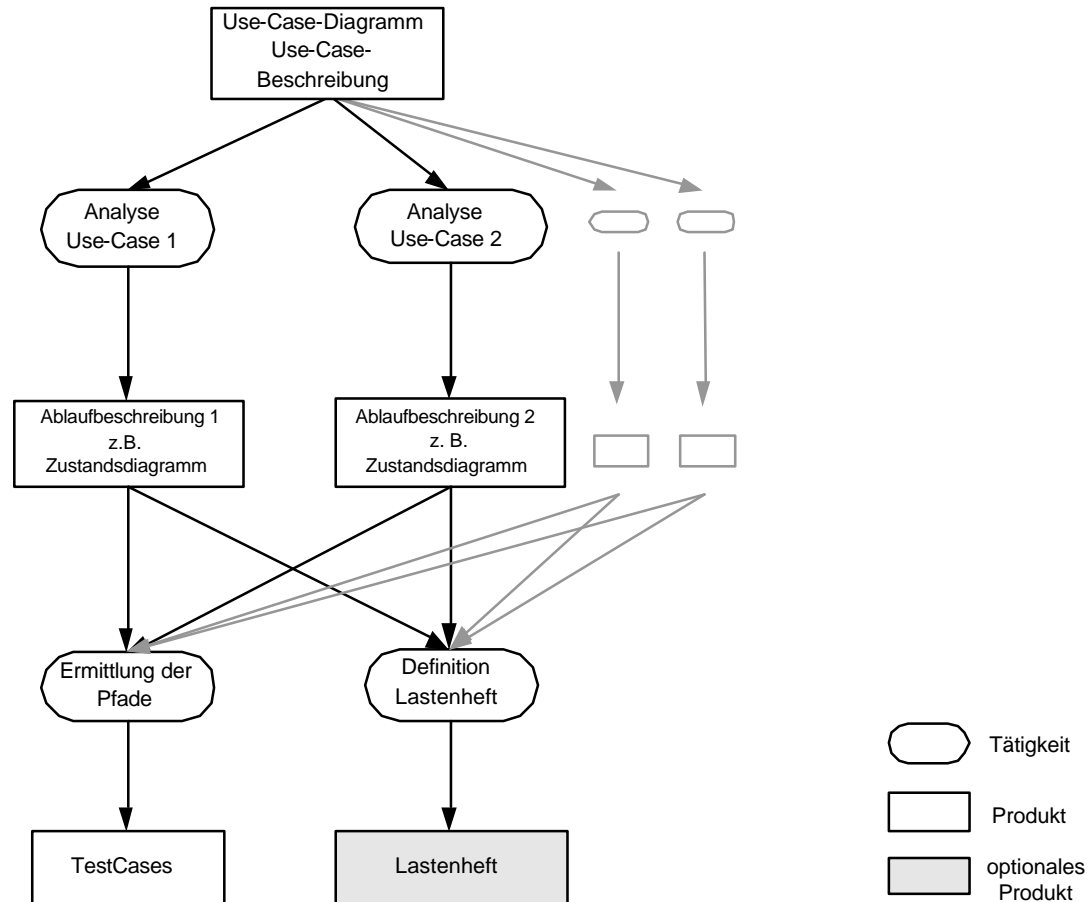
*zu anderen Use-Cases nicht
vollständig*

Neue Sichtweise – System Use Cases

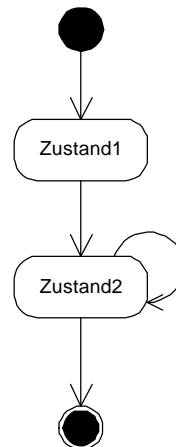


- > Kontextabgrenzung unerlässlich
- > Nachbarsysteme werden identifiziert (Schnittstellen)
- > Kontext legt Kommunikation (Events) fest
- > Use Cases werden durch Events gefunden
- > Essentielle Schritte beschreiben generelle Funktion
- > Ausnahmefälle werden „grob skizziert“
- > Glossar/Wertedefinition wird festgehalten

Use-Case → Test-Case: Ein Überblick



Ablaufbeschreibung: Aktivitäts- / Zustandsdiagramme



Scope

- Darstellung von zeitlichen und logischen Abhängigkeiten

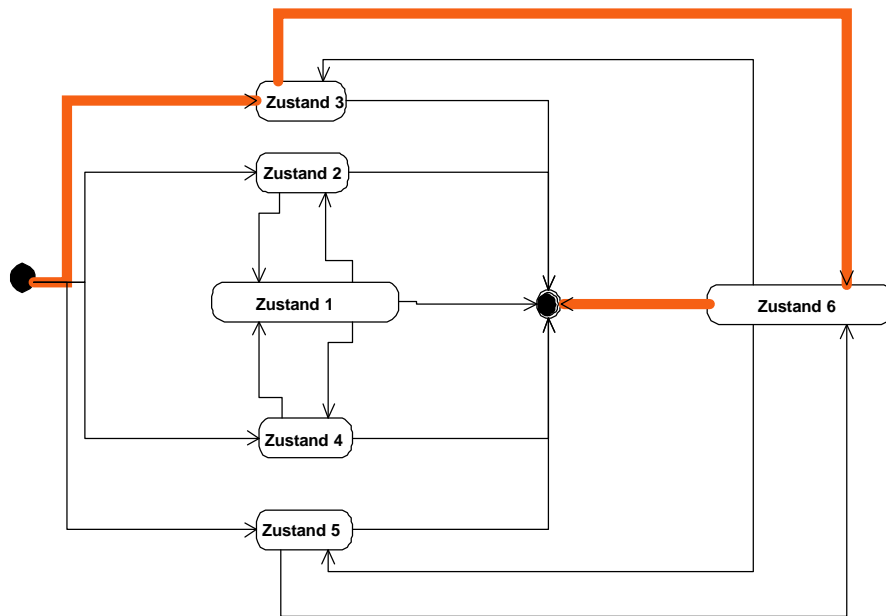
To do

- Ablaufbeschreibung weiter detaillieren
- Sonderfälle ergänzen

Benefit / Result

- Formale, eindeutige und systematisch erstellte Ablaufbeschreibung mit allen möglichen Abhängigkeiten

Ermittlung der Testpfade



Scope

- Extraktion der einzelnen Schritte eines Testfalles

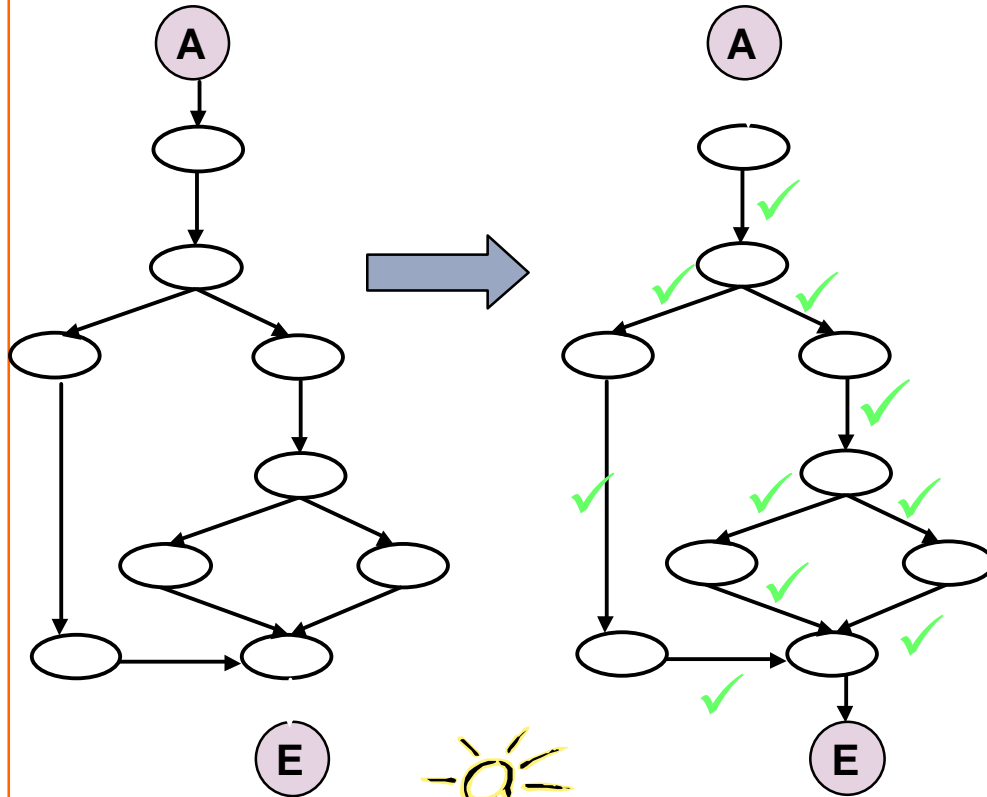
To do

- Alle Wege von Start- zum Endknoten im Diagramm ermitteln

Benefit / Result

- Beschreibung der Eingaben (Ereignisse des Benutzers)
- Beschreibung erwarteter Ausgaben (sichtbare Aktivitäten)
- Vollständige Testabdeckung

Betrachtung der Übergänge



Scope

- Minimierung des Testaufwandes

To do

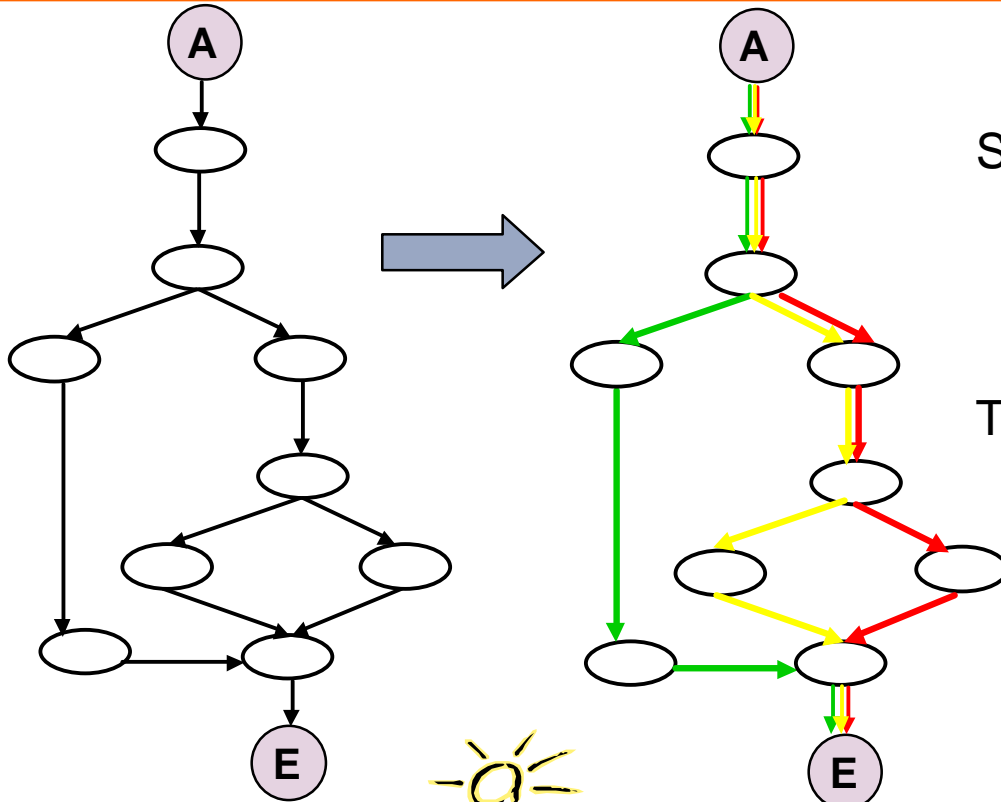
- Forderung: Jede Kante mind. einmal besuchen
- Suchalgorithmus anwenden

Benefit / Result

- Reduzierte Anzahl an Wegen, um die Testabdeckung zu erfüllen



Betrachtung der längsten Pfade



Scope

- Minimierung des Testaufwandes

To do

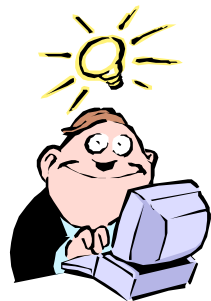
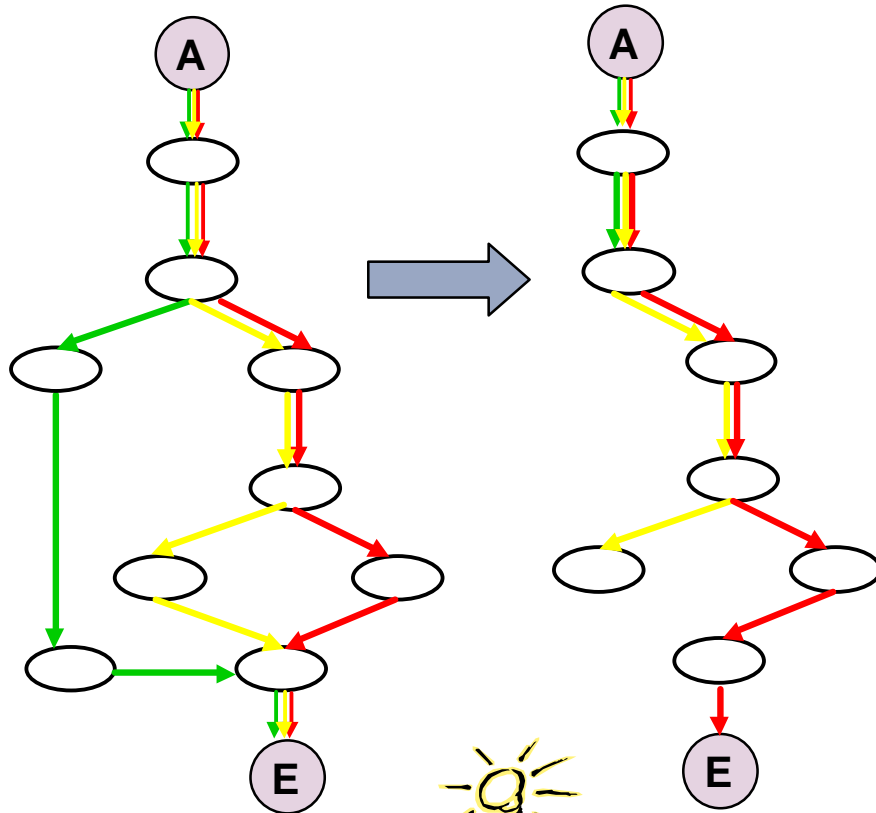
- Längsten Pfad suchen
- Markiere besuchte Übergänge
- Längsten Pfad suchen
- ...

Benefit / Result

- Minimale Anzahl an Wegen, um die Testabdeckung zu erfüllen



Reduktion durch Festlegen des Test-Scope



Scope

- Minimierung des Testaufwandes durch Vorgaben

To do

- Ermittlung der Ein- Ausgaben
- Auswahl der betrachteten Ein- Ausgaben durch Tester
- Automatische Ermittlung der Testpfade

Benefit / Result

- Minimale Anzahl an Wegen, um die Testabdeckung bzgl. des festgelegten Scopes zu erfüllen

Testfälle in Form gebracht

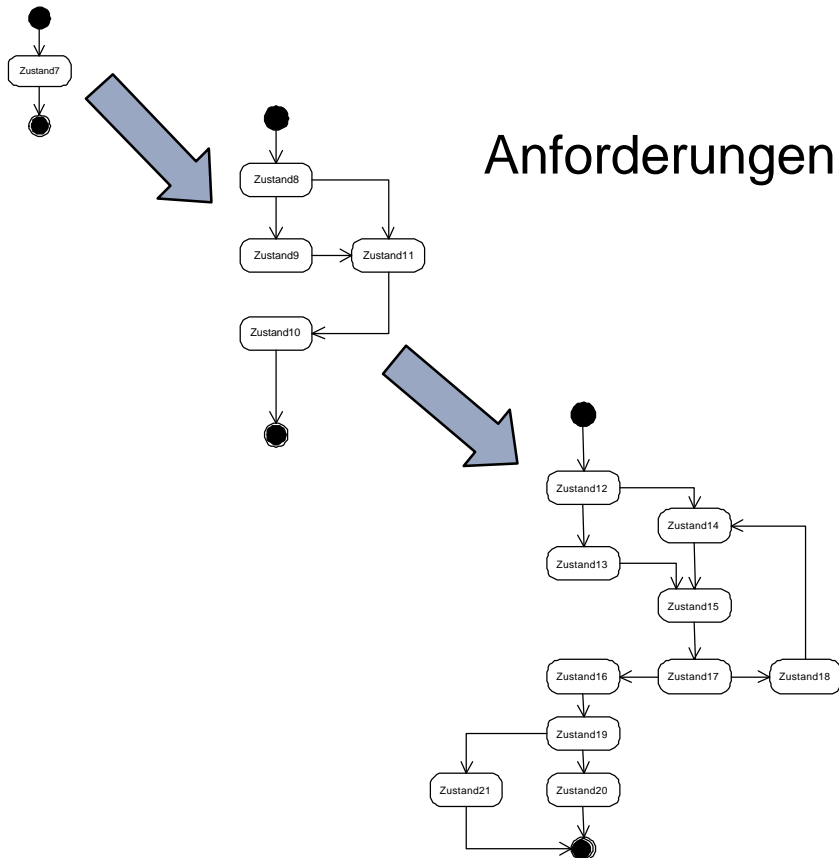
Template für die Testfalldokumentation



Testfall ID		Testfall Name				Getestetes System		
Zugehörige Use Cases								
Kurzbeschreibung / Testzweck								
Test Typ	Priorität		Test Suite			Autor		Datum
Vorbedingungen						Zugehörige Dokumente		
Nachbedingungen								
Schritt nr.	Zustand/ Aktivität	Ereignis / Exit Bedingung	Typ	Akteure	Interaktionskomponente	Erwartetes Ergebnis	Erfolg	Bemerkung

Vom Groben ins Feine

Verfeinerung der Ablaufbeschreibungen



Scope

- Schrittweise Entwicklung der Gesamtfunktionalität bis zur gewünschten Tiefe

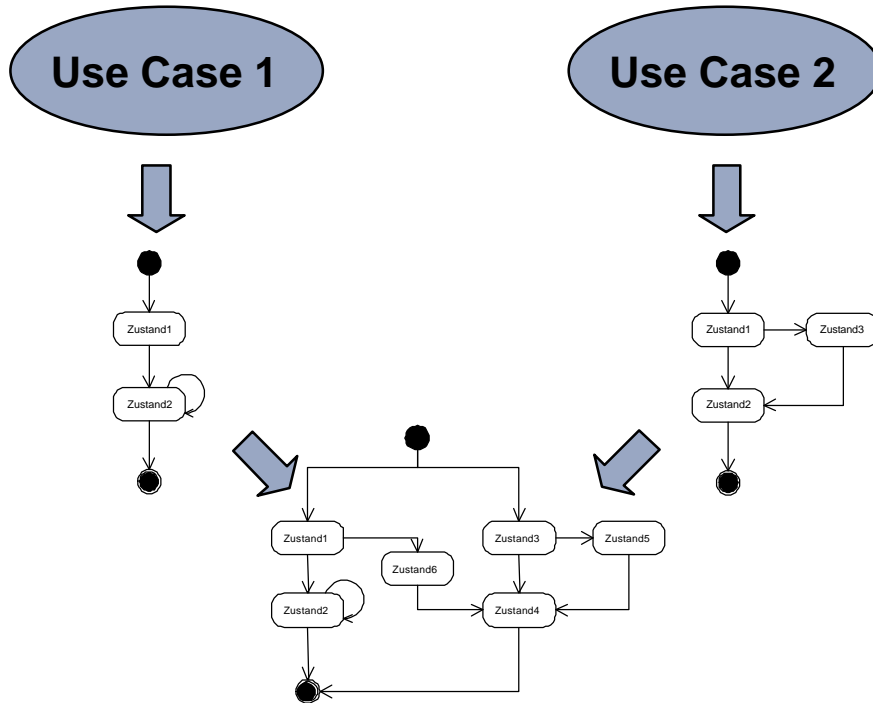
To do

- Iteratives ergänzen der Ablaufbeschreibungen

Benefit / Result

- Verschiedene Abstraktionsstufen
- Inkrementelle Verfeinerung der Beschreibungen

Komposition der Use Cases zum Systemprozess



Scope

- Komposition der Use Cases

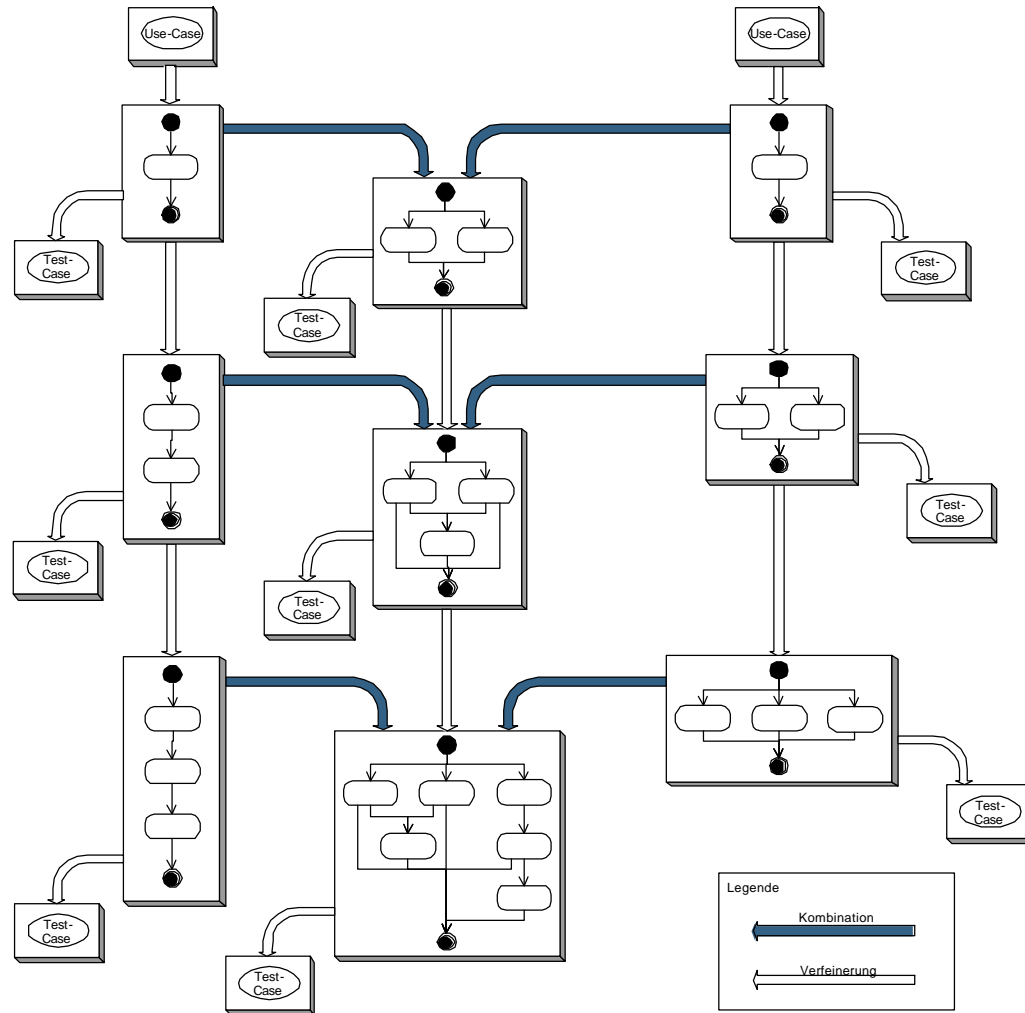
To do

- Einzelne Prozesse kombinieren

Benefit / Result

- Beschreibung des Gesamtsystemverhaltens
- Basis für Use Case übergreifende Testfälle

Kombination der Komposition und Verfeinerung





Zusammenfassung:

- Spezifikation und Test – Die zwei Seiten einer Medaille
- nachvollziehbare Kommunikation RE \leftrightarrow Test
- UC und Modelle erlauben automatische Testfallerzeugung

Weitere Arbeiten:

- Toolunterstützung in Arbeit
- Eingliederung in bekannte Vorgehensmodelle
 - Synergien ausnutzen
- Erweiterungen der UC und des Vorgehens auf Produktfamilien