



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN  
University of Applied Sciences

# Agiles Requirements Engineering *Priorisierung, UI-Design und Test*

Master Technische Informatik – Embedded Systems –  
Prof. Dr.-Ing. Hartmut Schirmacher

# Inhalt

- Priorisierung, Change Control und CCB
- Einbindung von Interaktionsdesign – wann und wie?
- Anforderungsbasiertes Testen
- Verfolgbarkeit und Änderungsmanagement



# Priorisierung und Change Management



# Priorisierung: Backlog und Releases

Prio ↑

ID	Beschreibung	Aufw.	Uns.
SN37-A	As a DBS nurse, ...	3d	2d
SN24-A	As a DBS nurse, ...	4d	0d
SS2000	As a service guy, ...	2d	1d
D4452	Defect: when entering...	2d	2d
SP2	As a patient, ...	6d	3d
SI44	As an investor, ...	8d	4d
...	...		
Release			
SP9	As a patient, ...	20d	10d
SI3-C	As an investor, ...	?	

↑

- Priorisierung bringt die Liste von Anforderungen / Stories in eine Reihenfolge
- dringendste / wertvollste Story zuerst
- Ein Release ist eine horizontale Grenze in der Liste: welche Stories gehören hinein, welche nicht
- Erste und wichtigste Frage: was muss **nicht dringend** in das nächste Release? → Verschieben hinter die Linie, nicht mehr damit beschäftigen
- Stories von oben nach unten nach und nach genauer schätzen und ggf. verfeinern
- Neben neuen Stories enthält das Backlog auch Defekte (Bugs)

Je weiter unten eine Anforderung steht, desto ungenauer darf die Abschätzung sein (Aufw. = Aufwand, Uns. = Unsicherheit)



# Wer priorisiert? Das Change Control Board (CCB)

Welches Wissen wird für Priorisierung benötigt?

- Bedeutung der Stories / Defekte für den Markt

- Welche Story hat welchen Wert
- Kano: Basis-, Leistungs- und Begeisterungsfaktoren
- Welcher Bug macht wie viele Probleme bei den Kunden

Product Owner,  
Vertreter aus Marketing  
und Support

- Bedeutung der Stories / Defekte für technischen Stakeholder

- Architektur, Abhängigkeiten in der Implementierungsreihenfolge
- Wert / Auswirkung von Refactorings
- Realisierbarkeit (klar genug formuliert? Aufwand?)

Vertreter der Entwicklung

- Anforderungen von Seiten der Qualitätsabteilung

- Als kritische einzustufende Defekte
- Einhaltung / Start der entsprechenden Eskalations-Prozesse  
z.B. bei Auftreten eines Defekts, der gemeldet werden muss

Vertreter der Q-Abteilung

CCB, regelmäßige  
Treffen zur Priorisierung  
neuer Defekte + Stories



# Beispiel DBS: Priorisierung einiger Stories



## Story 18: Kathode in Anode umwandeln

*Bei einer monopolarer Stimulation mit 2+ Kontakten möchte der User eine der beiden Kathoden in eine Anode umwandeln, um das Feld lokal zu begrenzen und damit Nebeneffekte zu unterdrücken.*

## Story 19: Anode hinzufügen

*Bei einer monopolarer Stimulation mit einem Kontakt möchte der User eine Anode hinzufügen, um das Feld lokal zu begrenzen und damit Nebeneffekte zu unterdrücken.*

Diskussion mit Don: er glaubt, das kommt öfter vor als Story 18

## Story 74: Automatische Amplitude für Anoden

*Der User möchte meistens, dass die Amplitude der Anoden sich automatisch aus der Summe der Kathoden berechnet, um den gesamten Strom über die Anoden abfließen zu lassen.*

Diskussion mit Don: für 1.7 reicht Story 74, Story 75 raus.



## Story 75: Manuelle Amplituden für Anoden

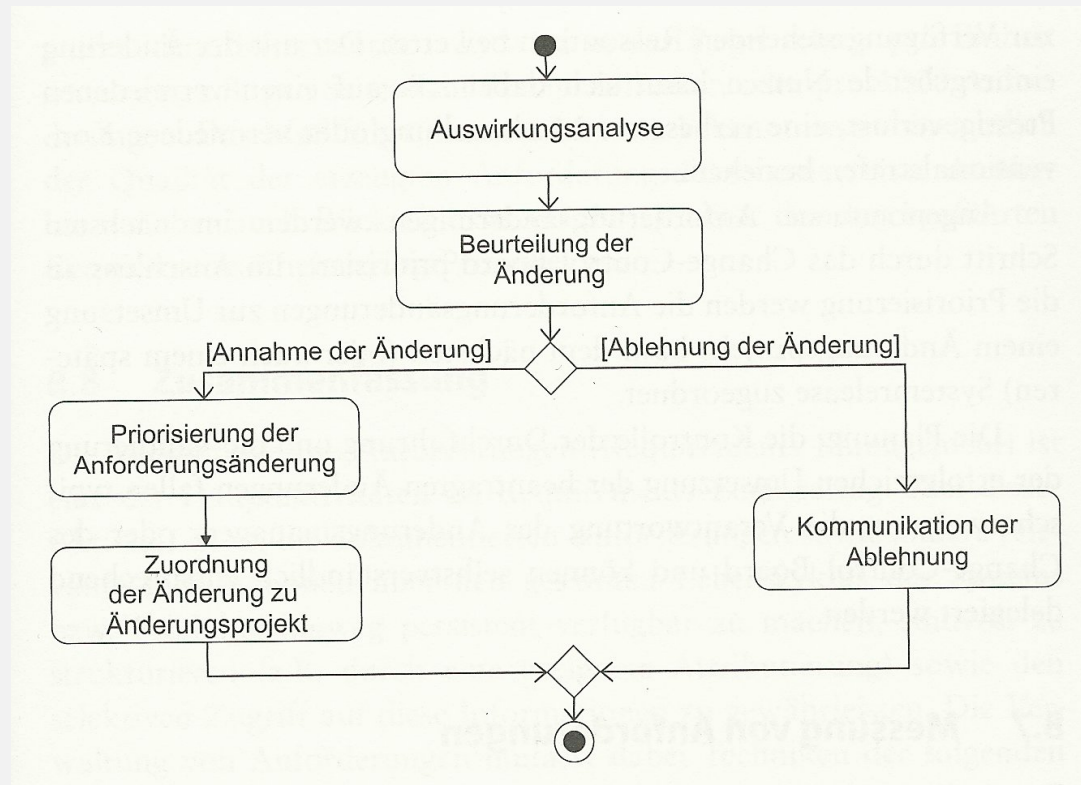
*Der User möchte in seltenen Fällen, dass die Amplituden der Anoden manuell definiert werden können. In diesem Fall fließt ggf. ein Teil des Stroms über das Gehäuse des IPG ab.*



# Software-Änderungen sind CCB-Sache

Kommunikation  
ggf. auch bei  
Priorisierung  
erforderlich!

Ansprechpartner  
im CCB wichtig für  
Teamkommunikation



# Einbindung von Interaktionsdesign





# Wann kommt das Interaktions-Design?

- Je mehr Stories / Szenarien ich kenne, desto besser kann ich entscheiden, wie mein UI strukturiert sein muss
  - Welche Benutzer gibt es, wie unterscheiden sie sich?
  - Welche Situationen gibt es, und was will der Benutzer genau erreichen?
  - In welchen Begriffen denkt der Benutzer?
  - Welche Operationen kommen wie häufig vor?
  - Welche Operationen werden aus unterschiedlichen Situationen heraus oder mit unterschiedlichen Zielen ausgeführt?
  - Von welcher Operation soll zu welcher anderen Operation navigiert werden können?



# Wann kommt das Interaktions-Design?

Einerseits:

- Je mehr Stories / Szenarien ich kenne, desto besser kann ich entscheiden, wie mein UI strukturiert sein muss

Andererseits:

- Sobald ich einen Screen entwerfe, können sich alle Stakeholder die Szenarien viel besser vorstellen, und man kann viel konkreteres Feedback einholen

Aber...

- man legt sich ggf. zu früh auf bestimmte Strukturen / Elemente fest
- man hält sich mit jeder Menge Details auf



# Einbindung des Interaktions-Designs

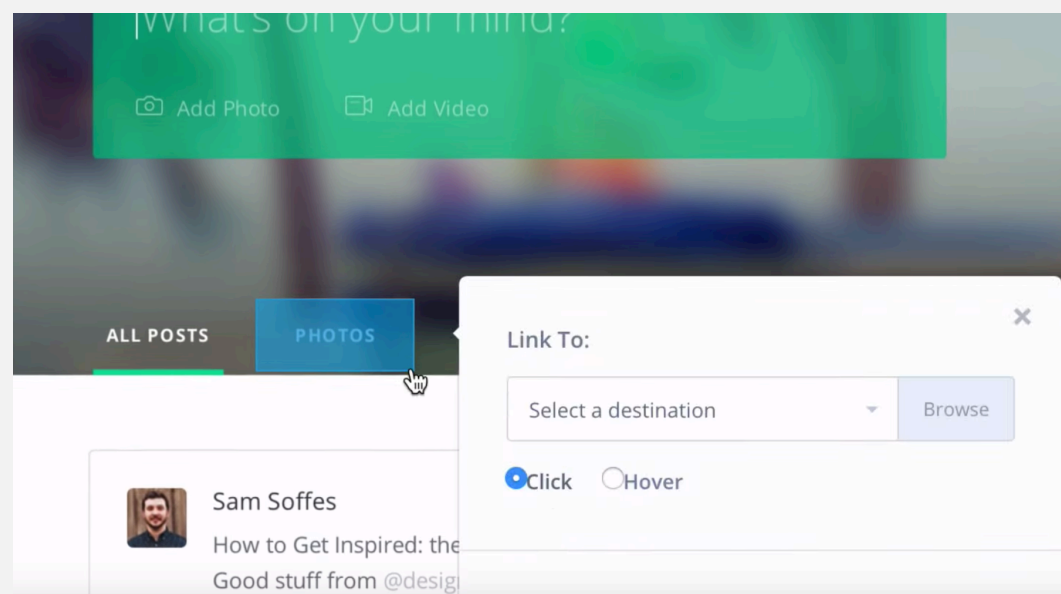
- Sammeln und priorisieren der wichtigsten Szenarien
- Workshop mit PO, Designer, Entwicklung
  - Durchsprechen der Stories für das Release im Zusammenhang
  - Gemeinsames Verständnis, Überblick
  - Gemeinsame Liste von Fragen an RE / Stakeholder
- Design-Sitzungen mit PO + Designer
  - Entwurf der Screens, Interaktionen, Navigation
  - Hierbei Verfeinerung der Szenarien, Ergänzung durch State Charts etc.
- Review der UI-Designs mit Stakeholdern (z.B. bei Sprint Review)
  - Auf Basis von Screens (Photoshop, Sketch) und Click Dummies (InVision)
  - Management, Kunden, Entwicklung
  - Nach Review: ggf. Änderungen am Design, RE-Review, ...
- Implementierung
  - nachdem Design mit den relevanten Stakeholdern abgestimmt ist



# Click Dummies

## Einfachster Workflow:

- Screens in verschiedenen Zuständen „malen“ (in Apps wie PhotoShop, Sketch)
- Auf jedem Zustand Hotspots markieren (in Apps wie InVision)
- Jeden Hotspot mit einem anderen Zustand verlinken



## Fortgeschrittene Features bei Click Dummies:

- Screen-Element in mehreren Ebenen definieren (z.B. Popups, die immer wieder auftreten)
- Templates und wiederkehrende Elemente
- ...



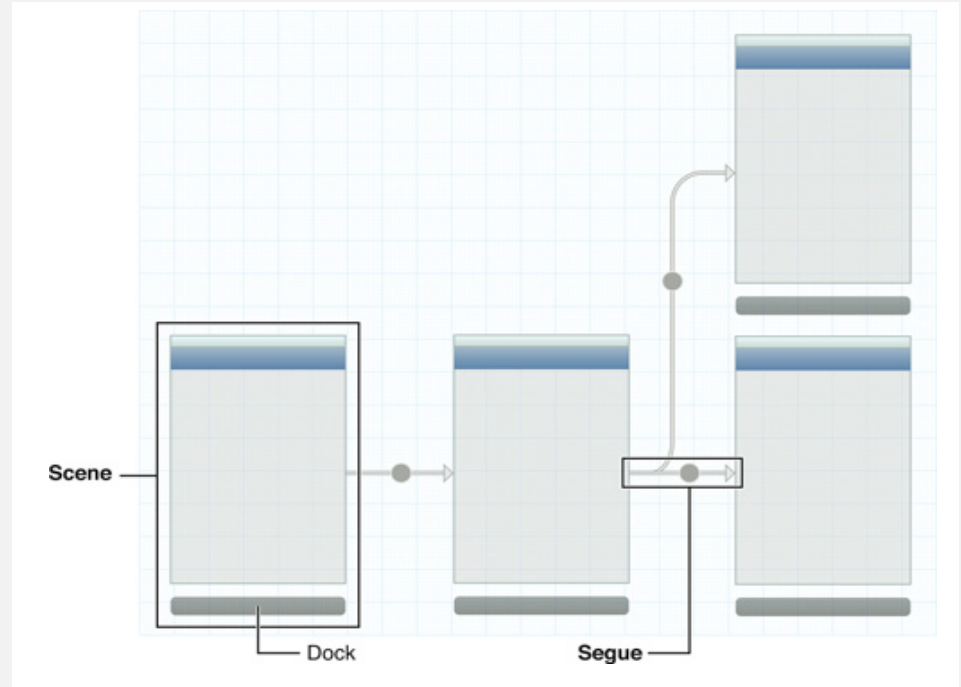
# Click Dummies im Einsatz

- Click Dummies können sehr gut mit verschiedensten Stakeholdern durchgegangen werden
  - Visuelle Darstellung der Funktionen erleichtert die Kommunikation
  - InVision erlaubt z.B. Kollaboration, Kommentierung, etc.
- Erlauben sehr frühzeitiges sehr konkretes Feedback
  - Anmerkungen zu visuellem Material oftmals sehr viel konkreter als zu textuellen Beschreibungen (Bild im der Vorstellung des Stakeholders nur schwammig)
- “pixel perfect“ vs. „wireframe“
  - Viele Stakeholder können nicht von visuellem Design abstrahieren
  - Daher oftmals Schwierigkeiten, wenn Click Dummy noch nicht “hübsch“
  - Falls nicht *pixel perfect*, dann deutlich als *nur wireframe* kennzeichnen, z.B. Tools wie *Balsamic* oder einfache Handzeichnung



# Storyboards (Apple / iOS)

A storyboard is a visual representation of the user interface of an iOS application, showing screens of content and the connections between those screens. A storyboard is composed of a sequence of scenes, each of which represents a view controller and its views; scenes are connected by segue [dt. *Übergang*] objects, which represent a transition between two view controllers.



Je nach Teamstruktur und Vorlieben des Designers können Click Dummies auch gleich in Form von Storyboards gebaut und dann direkt im Produkt verwendet werden.



# Anforderungsbasiertes Testen



# Softwaretest

- Definition nach Ernst Denert\*:

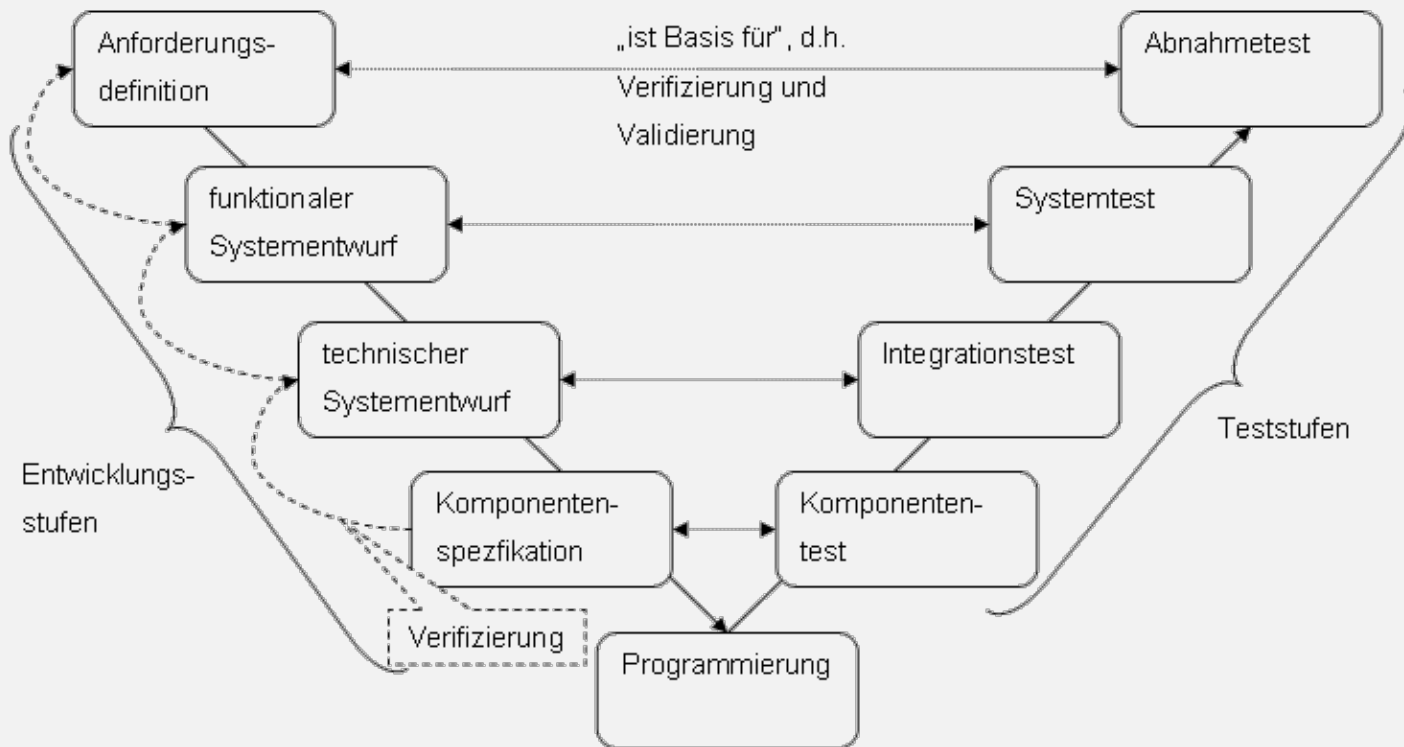
*[...] der überprüfbare und jederzeit wiederholbare Nachweis der Korrektheit eines Softwarebausteines relativ zu vorher festgelegten Anforderungen.*

- Norm: ISO/IEC/IEEE 29119 *Software Testing*





# Teststufen nach V-Modell \*



\*) es existiert eine Unzahl von Varianten dieser V-Grafik in der Literatur.

Bildquelle: <https://blog.milsystems.de/>



# Begriffe beim Testen \*

- Test Plan (Testplan)
  - Definiert Prozess und Strategie, definiert Abdeckung
  - Risikobasierte Definition, welcher Aspekt wie intensiv getestet werden soll
  - Festlegung von Testumgebung und –Daten
- Test Case (Testfall)
  - Spezifikation eines zusammenhängenden Tests; Verweise auf Anforderungen
- Test Protocol (Testablaufspezifikation)
  - Genaue Beschreibung der Schritte eines Testfalls mit erwarteten Ein- und Ausgaben
- Test Record (Testprotokoll ☹ **Verwirrung mit Test Protocol**)
  - Aufzeichnung dessen, was bei einer Durchführung einer Testfalls tatsächlich geschehen ist (schrittweise Ein- und Ausgaben, Vorfälle)



# Testabdeckung und Risikobetrachtung

- Im Testplan wird festgelegt, welche Komponenten und Funktionen des Produkts wie stark durch Tests abgedeckt sein müssen.
- Z.B.
  - Sämtlicher Code muss zu 100% durch Unit-Tests abgedeckt sein
  - Jede Schnittstellenanforderung muss durch mindestens einen Integrationstest abgedeckt sein
  - Jedes Benutzer-Szenario muss mindestens durch einen UI-Test (manuell oder automatisiert) abgedeckt werden
- Bei sicherheitskritischen Anwendungen ist diese Festlegung i.d.R. Risiko-basiert, d.h.
  - Risikoanalyse: was könnte wie wahrscheinlich passieren, was kann die Folge sein, wie können wir dieses Risiko mindern (risk mitigation)
  - Risiko für jede Hauptfunktion und jede technische Komponente analysieren
  - Aufwändigeres Testen für risikobehaftete Funktionen / Komponenten



## Beispiel: Test eines Szenarios



# Beispiel: Anforderungen

**Szenario UC7.4: Benutzer ermittelt die Amplitude für eine Anode, bei welcher der klinische Effekt der Therapie optimal ist**

Vorbedingungen: CP ist mit IPG verbunden; Patient ist gültig konfiguriert; CP befindet sich in Programmiersicht; alle Kontakte deaktiviert; Stimulation ist aus.

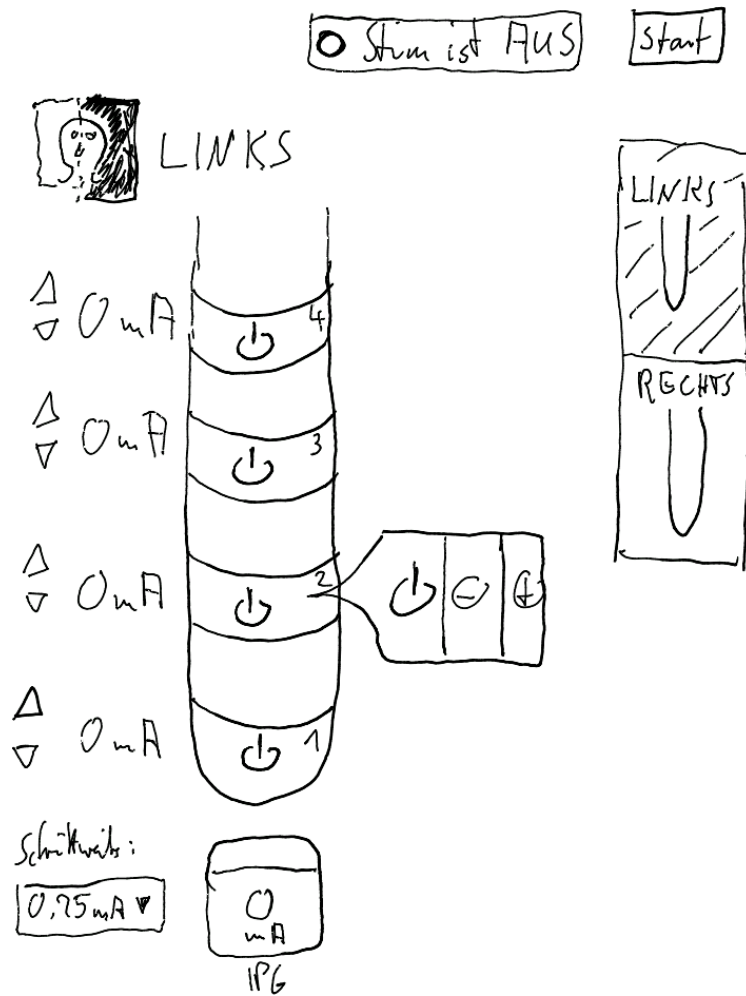
1. Benutzer wählt einen Kontakt auf einem Lead aus
2. Benutzer definiert den Kontakt als Anode
3. CP definiert die Amplitude des Kontakts initial als 0 mA
4. Benutzer schaltet Stimulation ein und erhöht schrittweise die Amplitude, bis erste Effekte am Patienten beobachtbar sind
5. Benutzer erhöht schrittweise weiter die Amplitude, bis zum Auftreten unerwünschter klinischer Effekte
6. Bei Bedarf verringert der Benutzer ggf. wieder die Amplitude um einen oder mehrere Schritte
7. Benutzer dokumentiert Kontakt, Amplituden und Effekte (z.B. auf Papier).

**Anforderung FA-034: Der Benutzer soll in der Lage sein, Amplituden von 0 bis 10 mA in Schritten von 0.1, 0.25, und 0.5 mA einzustellen.**

**Anforderung FA-007: Der Benutzer soll während der Programmiersitzung jederzeit in der Lage sein, die laufende Stimulation sofort zu stoppen.**



# Beispiel: UI-Wireframe



Dieses Beispiel dient nur der Illustration – es ist kein sehr guter Entwurf für die Übungsausgabe.

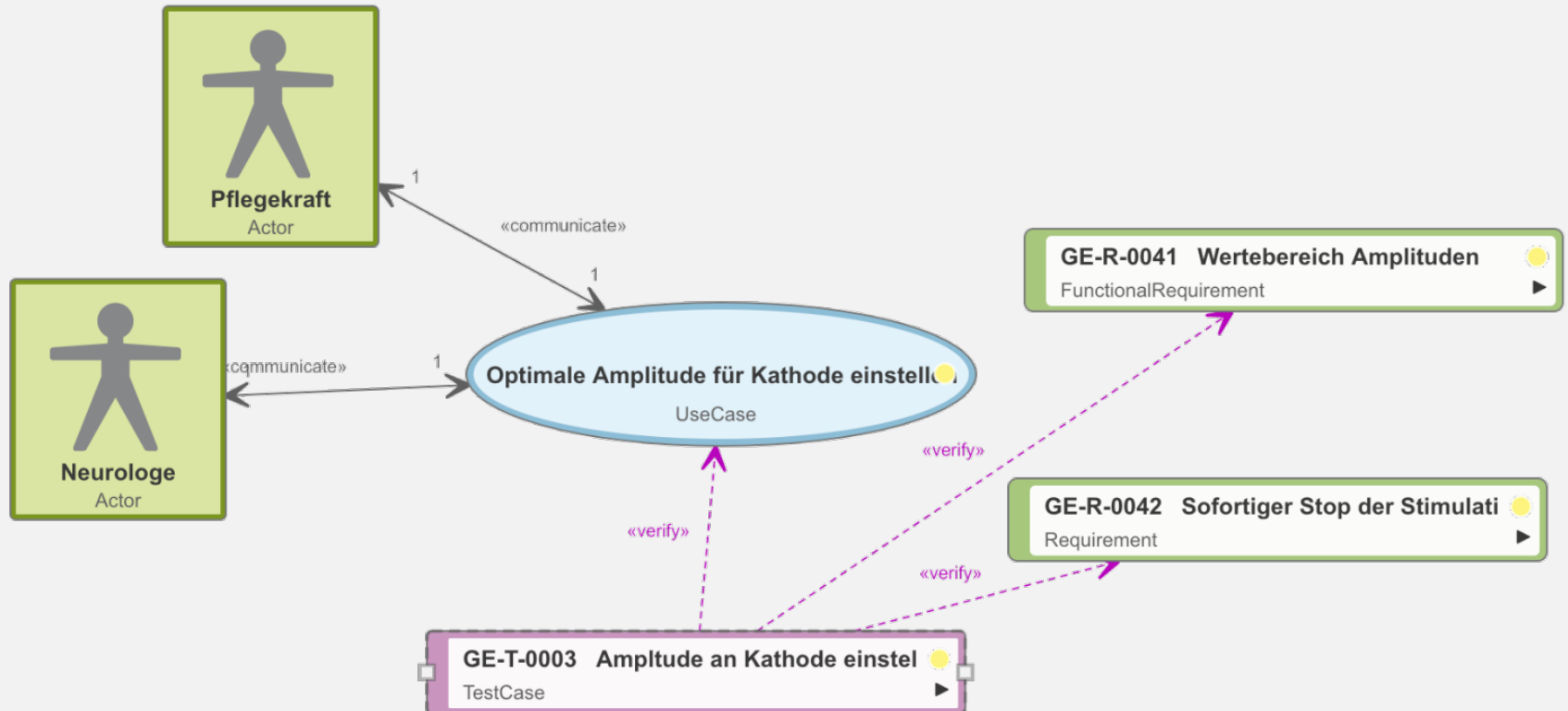


# Beispiel: Testfall

Testfall T012	Einstellung einer Anode
Typ	Integrationstest
Komponenten	CP, RC/IPG Simulator
Testumgebung	CP verbunden mit RC/IPG Simulator
Relevante Anforderungen	<ul style="list-style-type: none"><li>• UC7.4: Benutzer ermittelt die Amplitude für eine Anode, bei welcher der klinische Effekt der Therapie optimal ist</li><li>• FA-034: Der Benutzer soll in der Lage sein, Amplituden von 0 bis 10 mA in Schritten von 0.1, 0.25, und 0.5 mA einzustellen.</li><li>• Anforderung FA-007: Der Benutzer soll während der Programmiersitzung jederzeit in der Lage sein, die laufende Stimulation sofort zu stoppen.</li></ul>
Bemerkungen	<ul style="list-style-type: none"><li>• Die in UC7.4 beschriebene Interaktion außerhalb des CP werden hier nicht geprüft (Patientenreaktionen, Papierdokumentation).</li><li>• Der einstellbare Maximalwert für Amplituden wird hier nicht geprüft.</li></ul>



# Verknüpfter Testfall in objectiF RPM





# Beispiel: Testablauf (Test Protocol)

Durchzuführender Schritt	Erwartetes Ergebnis	Auswahl / Ergebnis	Pass / Fail / ?
Vorbedingungen: CP ist mit IPG verbunden; Patient ist gültig konfiguriert; CP befindet sich in Programmieransicht; alle Kontakte deaktiviert; Stimulation ist aus.	CP: Alle Kontakte zeigen „aus“-Symbol, IPG zeigt 0 mA, Stimulations-Indikator zeigt aus. Simulator: Stimulation ist aus		
1. Wählen Sie rechts einen Lead aus.	Ausgewählter Lead wird hervorgehoben und vergrößert dargestellt.		
2. Tippen Sie auf einen Kontakt und konfigurieren Sie ihn als KATHODE (-)	CP: Kontakt zeigt „-“ und 0 mA		
3. Wählen Sie eine Schrittweite von 0.25 oder 0.5 mA	CP: Gewählte Schrittweite wird hervorgehoben.		
4. Schalten Sie die Stimulation mit der Schaltfläche „Start“ ein	CP: Grüner Indikator, Text „Stim is AKTIV“		
	Simulator: Stimulation ist aus		
5. Erhöhen Sie die Amplitude schrittweise bis auf einen Zielwert zwischen 2 mA und 4 mA	CP: Amplitude wird um die ausgewählte Schrittweite erhöht und angezeigt		
	Simulator: genau ein Kontakt ist aktiv und zeigt ein Signal mit der im CP angezeigten Amplitude		
...	...		



# Beispiel: Test Record

Produktversion(en): CP 1.7RC4, Sim 1.7RC1  
 Testzeitpunkt: 2017-05-08 12:30 (GMT+1)  
 Tester: Hugo Hurtig

Durchzuführender Schritt	Erwartetes Ergebnis	Auswahl / Ergebnis	Pass / Fail / ?
Vorbedingungen: CP ist mit IPG verbunden; Patient ist gültig konfiguriert; CP befindet sich in Programmieransicht; alle Kontakte deaktiviert; Stimulation ist aus.	CP: Alle Kontakte zeigen „aus“-Symbol, IPG zeigt 0 mA, Stimulations-Indikator zeigt aus. Simulator: Stimulation ist aus		PASS
1. Wählen Sie einen Lead aus.	Ausgewählter Lead wird hervorgehoben und vergrößert dargestellt.	Lead L-STN	PASS
2. Tippen Sie auf einen Kontakt und konfigurieren Sie ihn als KATHODE (-)	CP: Kontakt zeigt „-“ und 0 mA	Kontakt 2 auf Lead	PASS
3. Wählen Sie eine Schrittweite von 0.25 oder 0.5 mA	CP: Gewählte Schrittweite wird hervorgehoben.	0.25	PASS
4. Schalten Sie die Stimulation mit der Schaltfläche „Start“ ein	CP: Grüner Indikator, Text „Stim is AKTIV“		PASS
	Simulator: Stimulation ist aus	Grüner Indikator an, aber kein Kontakt aktiv	?
5. Erhöhen Sie die Amplitude schrittweise bis auf einen Zielwert zwischen 2 mA und 4 mA	CP: Amplitude wird um die ausgewählte Schrittweite erhöht und angezeigt	Bis 2.25 mA	PASS
	Simulator: genau ein IPG-Kontakt ist aktiv und zeigt ein Signal mit der im CP angezeigten Amplitude	Kontakt 6 am Lead	PASS
...	...		



# Beispiel: weitere Tests

**Anforderung FA-034: Der Benutzer soll in der Lage sein, Amplituden von 0 bis 10 mA in Schritten von 0.1, 0.25, und 0.5 mA einzustellen.**

Hier testet man sinnvollerweise möglichst auch mit automatischen Tests, dass der Wertebereich im Modell vollständig eingestellt werden können (Unit-Tests), und dass die eingestellten Werte auch beim IPG (Simulator) ankommen (Integrationstests). Die UI-Tests beschränkt man üblicherweise auf wenige wichtige Fälle.

**Anforderung FA-007: Der Benutzer soll während der Programmiersitzung jederzeit in der Lage sein, die laufende Stimulation sofort zu stoppen.**

Das „jederzeit“ ist nicht wirklich testbar. Diese Anforderung sollte durch eine Vielzahl verschiedener Szenarien abgedeckt sein. Beispiel: in jeden möglichen Testfall zur Programmierung noch zwei Schritte “schalten Sie die Stim ab / schalten Sie die Stim wieder an” einbauen. Hier sollte der Testplan bereits eine höhere Abdeckung vorschreiben.



# Verfolgbarkeit von Anforderungen



# Verfolgbarkeit / Traceability

- Beziehungen zwischen Anforderungen, Spezifikationen und Tests
- Beziehungen zwischen Änderungen und allen obigen Elementen
- Versionierung
  - welche Version der Anforderung gehört zu welcher Version des Tests...

Einige wichtige Fragestellungen:

- Findet der Entwickler schnell heraus, welche Anforderung und Spezifikationen hinter einer geplanten Änderung stehen, und ob diese bereits abgestimmt / gültig sind?
- Aus welchen Anforderungen erschließt sich der Inhalt eines Tests?
- Durch welche Tests ist eine Anforderung abgedeckt?



# Nutzen der Verfolgbarkeit

- Nachweis, dass Anforderung auch vom System umgesetzt wird
- Identifikation von „Goldrandlösungen“
  - Goldrand = eigentlich gar nicht geforderte Eigenschaften, deren Realisierung das Budget belasten
- Auswirkungenanalyse
  - Auswirkung der Änderung eines Artefakts auf andere Artefakte
  - Automatische Veranlassung der Überprüfung
- Wiederverwendung
  - Modellierter Beziehung erleichtern es ungemein, Artefakte für die Wiederverwendung in anderen Projekten auszuwählen
- Zurechenbarkeit
  - Z.B. Zusammenrechnen aller Entwicklungsaufwände für eine Anforderung
- Wartung und Pflege
  - Unterstützt Identifikation von Ursachen und Auswirkungen (s.o.)



# Änderungsmanagement

Welche Auswirkungen hat eine Änderung

- Für die Risikoanalyse
- auf die Anforderungen und Tests
  - Stimmt eine Anforderung nicht (mehr)?  
→ Änderung Anforderung
  - Fehlt eine Anforderung?  
→ Hinzufügen Anforderung
  - Wird eine korrekte Anforderung nicht richtig umgesetzt?  
→ Überprüfung Spezifikation, Testplan und Tests
- Wurde eine Anforderung geändert: welche weiteren Anforderungen (z.B. Szenarien), Spezifikationen und Tests können betroffen sein?  
→ Prüfung, ggf. Änderung, erneute Freigabe der entsprechenden Artefakte
- usw.



Fragen?

