

- UCD Process (ISO 9241)

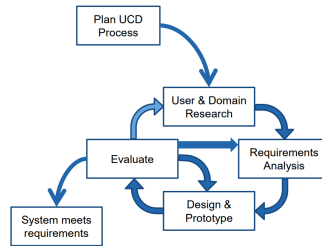


Abbildung 1: UCDPProcess

0.1 User & Domain Research

- **Ziele bez. Benutzer:**
 - Wer ist Benutzer
 - Was ist die Arbeit (Aufgaben, Ziele)
 - Wie sieht Arbeitsumgebung aus
 - Was wird gebraucht um Ziele zu erreichen
- Welche Sprache, Begriffe
- Normen (organisatorisch, kulturell, sozial)
- Pain Points (Brüche, Workarounds)
- Für mobile Apps:
 - Nutzungskontext
 - * Wo wird App benutzt (Umgebung)
 - * Wann wird App benutzt (Tageszeit, involvierte Personen, Randbedingungen)
 - * Warum wird App benutzt (Nutzen, Motivation, Trigger)
- **Ziele bez. Domäne:**
 - Business der Firma verstehen
 - Domäne verstehen (Sprache, Wichtigste Konzepte, Prozesse)

0.1.1 GUI Design Process

Methoden User & Domain Research

- Contextual Inquiry
 - Was ist das?(Beobachtung, Interview)
 - Was braucht es dazu? (Für den inquiry, z.B Videogerät)
- Interviews
- Beobachtung
- Fokusgruppen
- Umfragen
- Nutzungsauswertung
- Desktop Research (Dokumentenstudium, Mitbewerber)

0.1.2 Wichtige Artefakte

- **Personas**
- **Usage-Szenarien**
 - Kurze Geschichte
 - * **Usage Szenarien**
 - aktuelle Situation
 - in User and domain research verwendet
 - * **Kontextszenarien**
 - Zukünftige gewünschte Situation
 - in Anforderungsanalyse verwendet
- **Mentales Modell**
- **Domänenmodell**
- **Stakeholder Map**

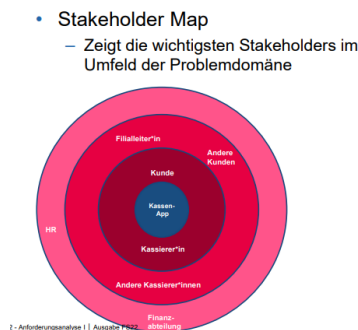


Abbildung 2: Stakeholdermap

- **Service Blueprint/Geschäftsprozessmodell**

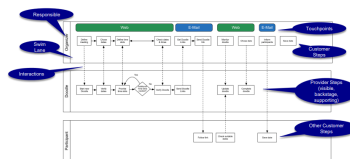


Abbildung 3: Blueprint

0.2 Anforderungsanalyse

Ziel:

- Ausgehend von den Resultaten des UCD -> User-Anforderungen ableiten:
 - Funktionale Abläufe, Interaktionen
 - * **Kontextszenarien**
 - * **Storyboards**
 - * **UI-Skizzen**
 - * **Use cases**
 - Konzepte, Beziehungen, Quantitäten
 - * **Kontextszenarien**
 - * **FURPS-Modell** (Functionality, Usability, Reliability, Performance, Supportability)

0.2.1 Use Cases

- **Akteur**
 - Primärakteur
 - Unterstützender Akteur
 - Offstage-Akteur
- **Keine Kann-Formulierungen**
- **3 Ausprägungen:**
 1. Kurz
 - Titel + 1 Absatz (Standardablauf)
 2. Informell
 - Titel + Informelle Beschreibung (können mehrere Absätze sein, beschreibt auch Varianten)
 3. Vollständig
 - Titel + alle Schritte und alle Varianten im Detail
 - UC-Name
 - Umfang
 - Ebene
 - Primärakteur
 - Stakeholders und Interessen
 - Vorbedingungen
 - Erfolgsgarantie/Nachbedingungen
 - Standardablauf
 - Erweiterungen
 - Spezielle Anforderungen
 - Liste der Technik und Datavariationen
 - Häufigkeit des Auftretens
 - Verschiedenes
- **Notation = Nomen + Verb**

0.2.2 Wie schreibt man Use Cases

Brief UC:

- Kurze Beschreibung des Anwendungsfalls in einem Paragraph
 - Nur Erfolgszenario
 - Sollte enthalten:
 - * Trigger des UCs
 - * Akteure
 - * summarischen Ablauf des UCs
 - Zu Beginn der Analyse

Casual UC:

- Informelle Beschreibung des Anwendungsfalls in mehreren Paragraphen
 - Nur Erfolgszenario + wichtigste Alternativszenarien
 - Sollte enthalten:
 - * Trigger des UCs

- * Akteure
- * Interaktion Akteurs mit System
- Zu Beginn der Analyse

Fully-dressed UC:

- Detaillierte Beschreibung des Ablaufs mit allen Alternativszenarien
- Ende der Inception und v.a. in Elaborationphase
- Die wichtigsten UCs(10%), die die Architektur bestimmen

Formaler Aufbau:

1. UC-Name

- Aktiv formulieren (Verb + ev.Objekt)
- Beschreibt Job(Ziel,Aufgabe),den Akteur ausführen will

2. Umfang(Scope)

- Beschreibt das zu entwickelnde System (SuD= System under Development)

3. Ebene(Level)

- Anwenderziel oder
- Subfunktion

4. Primärakteur

- Hauptakteur des UCs
 - Primärer Nutzniesser des UC
 - initiiert den UC
 - Interagiert hauptsächlich mit dem System

5. Stakeholders und Interessen

- Für wen ist der UC sonst noch relevant und welche Interessen hat er daran?

6. Vorbedingungen(Preconditions)

- Was ist die unmittelbare Voraussetzung, damit UC ablaufen kann? (Nur die wichtigsten,offensichtliche Voraussetzungen)

7. Erfolgsgarantie/Nachbedingungen(Success Guarantee)

- Was muss gewährleistet sein nach erfolgreichem Ablauf von UC

8. Standardablauf (Main Success Scenario)

- Wichtigster Teil des UCs(Beschreibung erfolgreichen Ablaufs des UCs)
- Beschreibung Interaktion Primärakteurs mit dem System(+allenfalls Interaktion mit unterstützenden Akteuren)
- Startpunkt= nach den Vorbedingungen
- Keine Lösungsdetails

9. Erweiterungen (Extensions)

- Alternative Erfolgs- aber auch Misserfolgsszenarien
- Erweiterungen / alternative Abläufe
- * = Alternativablauf zu jeder Zeit auftreten kann
- Interaktion des alternativen Ablaufs analog zu Hauptszenario

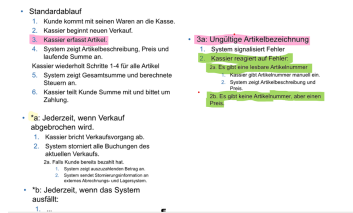


Abbildung 4: Beispiel-Fully-Dressed.

- **Spezielle Anforderungen (Special Requirements)**
 - Weitere Anforderungen, die aus diesem UC resultieren
- **Liste der Technik und Datavariationen (Technology and Data Variations)**
 - Alternative I/O-Methoden, Datenformate, etc.(Notation wie bei Erweiterung)
- **Häufigkeit des Auftretens(Frequency of Occurance)**
 - Wie häufig tritt dieser UC auf?
- **Verschiedenes (Miscellaneous)**
 - Offene Fragen/ Probleme

von Anwendungsfällen zu konkreten Funktionalitäten:

- Systemsequenzdiagramme
- Operation Contracts

0.2.3 UML Sequenzdiagramm (SSD)

Zeigt: Interaktion der Akteure mit dem System

- Welche Input-Events auf das System einwirken
- Welche Output-Events das System erzeugt

SSD können auch Interaktionen zwischen SuD und externen unterstützenden System zeigen.

Ziel: Wichtigste Systemoperationen identifizieren, die das System zur Verfügung stellen muss (API) für einen gegebenen Anwendungsfall

Wie Systemoperationen finden:

1. Szenario für UC Schritt für Schritt durchgehen
2. Für jeden Schritt Systemoperation überlegen
 - Geeigneten, präzisen Namen wählen (POV Akteur)
 - Welche Info braucht das System um Systemop. auszuführen? (Wenn nicht vorhanden im System -> Parameter)

0.2.4 Operation Contract

Definition: Spezifiziert (System)Operation

- Name plus Parameterliste
- Vorbedingung (Was muss zwingend erfüllt sein damit Systemoperation aufgerufen werden kann)
- Nachbedingung
 - Was hat sich alles geändert nach Ausführung (Erstellte / gelöschte Instanzen, Assoziationen, geänderte Attribute), Im Präsens schreiben
 - basierend auf Domänenmodell

Wann Operation Contracts?

- Nur wenn aus Anwendungsfall nicht klar wird, was Systemoperation genau machen muss
 - Meist nur bei sehr komplizierten Operationen und/oder
 - Wenn Entwicklung der Systemoperation ausgelagert wird
- Erst gegen Ende des Meilensteins Lösungsarchitektur oder kurz vor Start des Designs der Systemoperation

0.2.5 Zusätzliche Anforderungen

- weitere Funktionale (grosser Teil schon von UCs beschrieben)
- Nicht-Funktionale

Formulierung

- Anforderungstatements
 - Als Anforderung formuliert
 - messbar/verifizierbar
- So wenig wie nötig
 - Nur diejenige, die begründet gefordert werden
 - Keine ersten Lösungsideen als Forderungen
- User-Stories
 - In einem Satz wer,was,warum fordert
 - Erfüllen einige Bedingungen automatisch
 - die anderen Bedingungen (Messbarkeit etc.) sollten auch erfüllt sein

Checkliste **FURPS+**

- **F**unctionality
 - Features, Fähigkeiten, Sicherheit
- **U**sability
 - Usability Anforderungen (Kap.5.3)
 - Accessibility
- **R**eliability

- Fehlerrate, Wiederanlauffähigkeit, Vorhersagbarkeit, Datensicherung
- **Performance**
 - Reaktionszeiten, Durchsatz, Genauigkeit, Verfügbarkeit, Ressourceneinsatz
- **Supportability**
 - Anpassungsfähigkeit, Wartbarkeit, Internationalisierung, Konfigurierbarkeit
- **+**
 - Implementation (HW,BS,Sprachen, Tests...)
 - Interface
 - Operations
 - Packaging
 - Legal

Glossar

- **Einfaches Glossar**
 - Begriffe im Projekt und SW-Produkt
 - beliebige Elemente
- **Data Dictionary**
 - Zusätzliche Datenformate, Wertebereiche, Validierungsregeln