

# Usability & User Experience Engineering

## *Einheit 1: Engineering Prozess*

Univ. Ass. Alexander Meschtscherjakov



# ORGANISATORISCHES

# Organisatorisches



- VO (Meschtscherjakov)
- Mi 8:00 -10:00 T02 (teilgeblockt) – dh Termine fallen aus.
- **2. Einheit: Mi 24.10.2018**
- Zugehöriges PS (Mirnig) – Do 8:00-10:00 – Start: Do 10.10.
- 033, 511 Bachelorstudium, Informatik
- Wahlmodul W3: HCI
- 999 Fakultätsübergreifende Lehre
- 2 ECTS (=50 Arbeitsstunden)
  - 21h VO (10 Termine a 120 Min)
  - Rest: Lektüre & Klausurvorbereitung
- Folien im Blackboard
- Klausur am Semesterende (Mischung aus Multiple Choice und Fragen)

# Inhalt



- Einführung in die Grundprinzipien des Usability Engineerings
- Einführung in die Grundbegriffe von Usability / User Experience
- Aktivitäten, Methoden, Werkzeuge in der Analyse
- Aktivitäten, Methoden, Werkzeuge im User Interface Design
- Aktivitäten, Methoden, Werkzeuge in der Evaluation

# Literatur

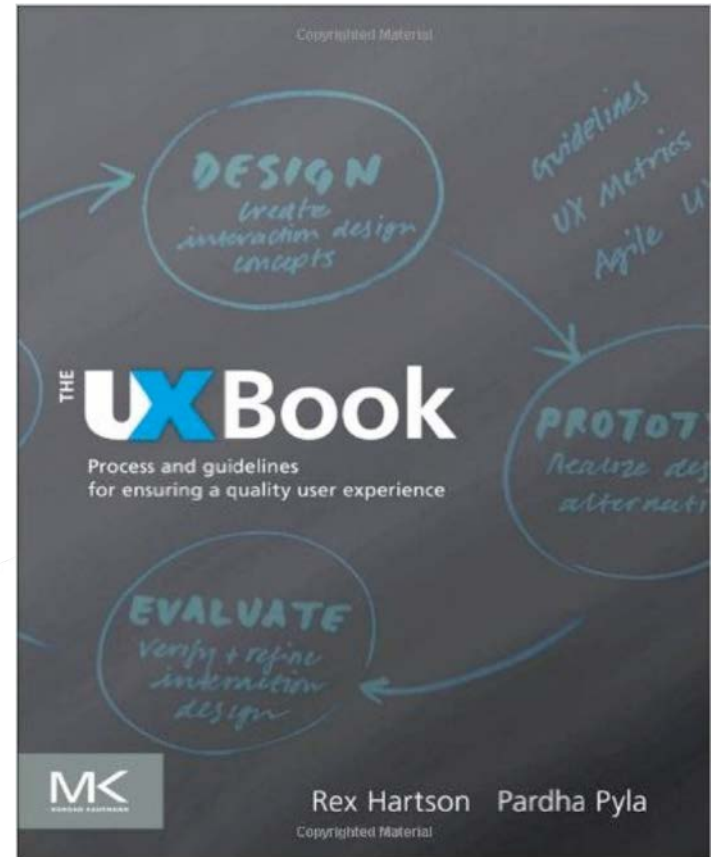


## The UX Book

Process and Guidelines for Ensuring a Quality User Experience

Rex Hartson & Pardha Pyla

<http://www.theuxbook.net>



# Was ist Usability?



# Usability: Benutzerfreundlichkeit



*The extent to which a product can be used by specified users to achieve specified goals with **effectiveness**, **efficiency** and **satisfaction** in a specified context of use.*

[ISO 9241-11]

# Was ist User Experience (UX)?





# User Experience: Nutzungserlebnis



*A person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service.*

*Note 1: User experience includes all the user's emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviors and accomplishments that occur before, during and after use.*

*Note 2: User experience is a consequence of brand image, presentation, functionality, system performance, interactive behavior and assistive capabilities of the interactive system, the user's internal and physical state resulting from prior experience, attitudes, skills and personality, and the context of use.*

[ISO DIS 9241-210]



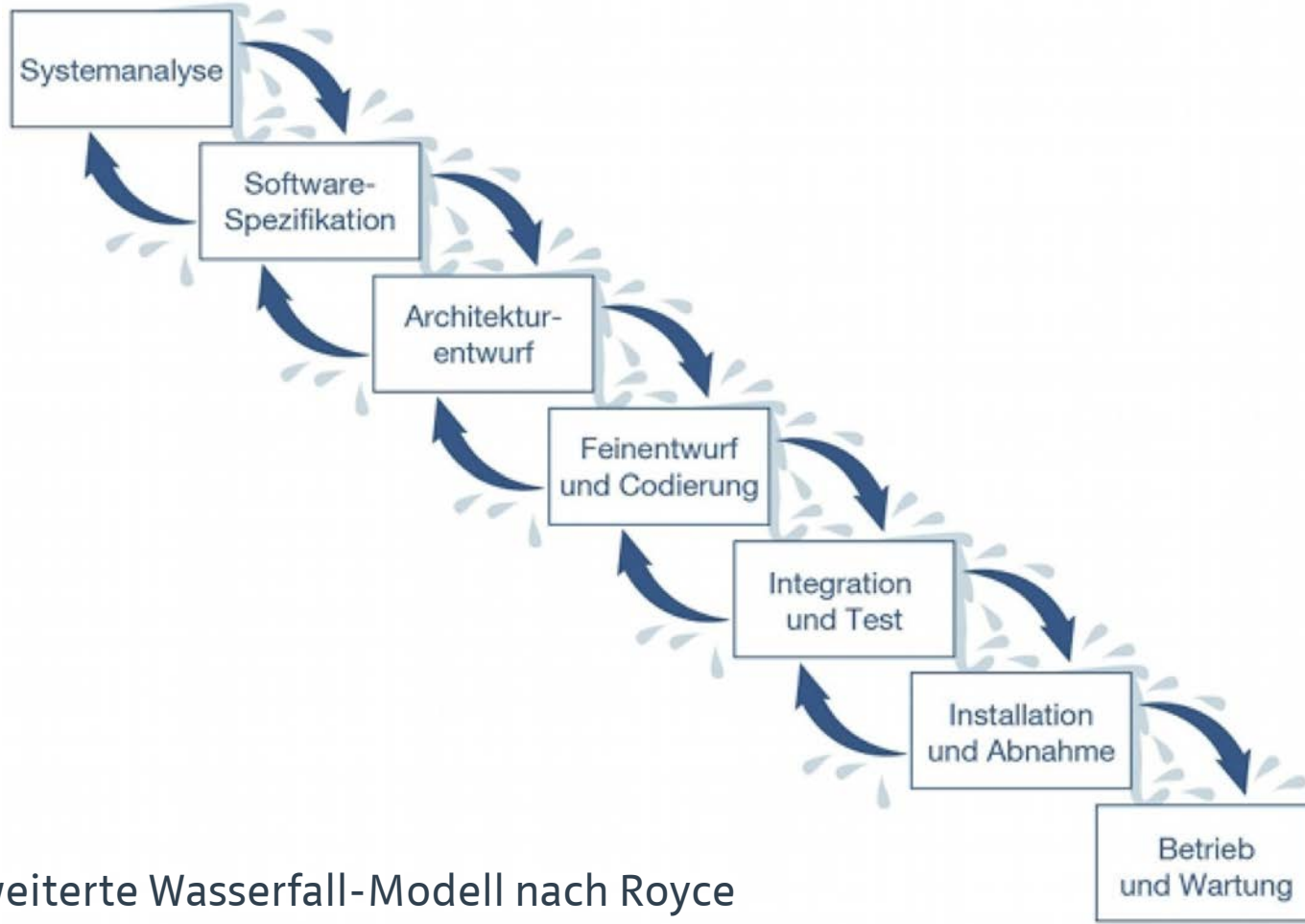
# ENGINEERING PROZESS

# Prozess Prinzipien



- Sicherstellen dass kein Aspekt der User Experience ohne expliziter, bewusster Absicht entsteht
- Jede mögliche Benutzeraktion betrachten die Benutzer wahrscheinlich unternimmt
- Alle Benutzererwartungen zu jedem Zeitpunkt verstehen
- Früher und andauernder Fokus
  - Auf Benutzer
  - Auf Aufgaben
  - Auf Benutzbarkeitsanforderungen
- Ziele
  - Was entwickeln, dass auch benutzt wird
  - Änderungen bevor diese zu teuer sind
  - Dokumentationen und Schulung einfacher
  - Reduzierte Wartung, Updates

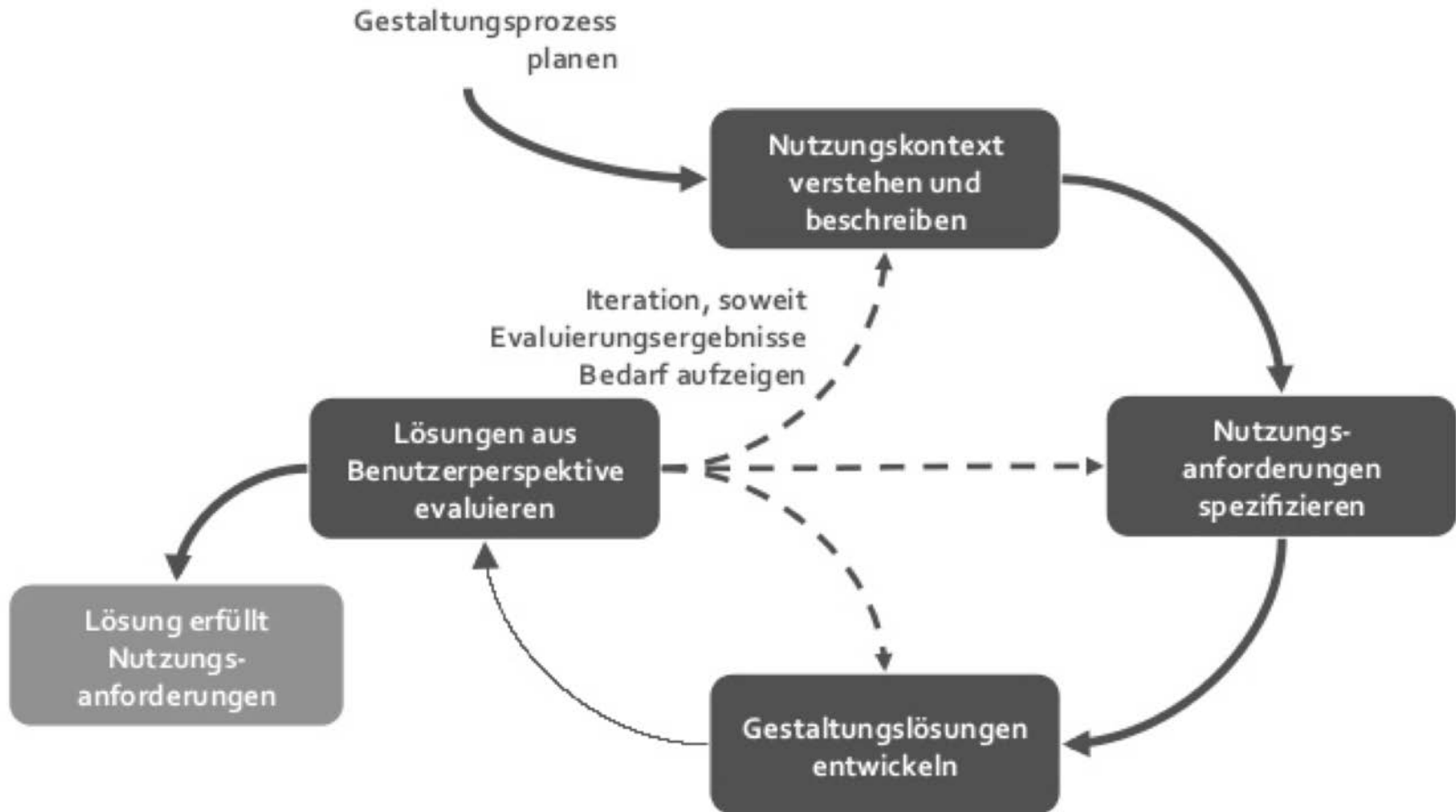
# Wasserfallmodell



Erweiterte Wasserfall-Modell nach Royce

# ISO 9241-210, 2010

## Human-centred design activities



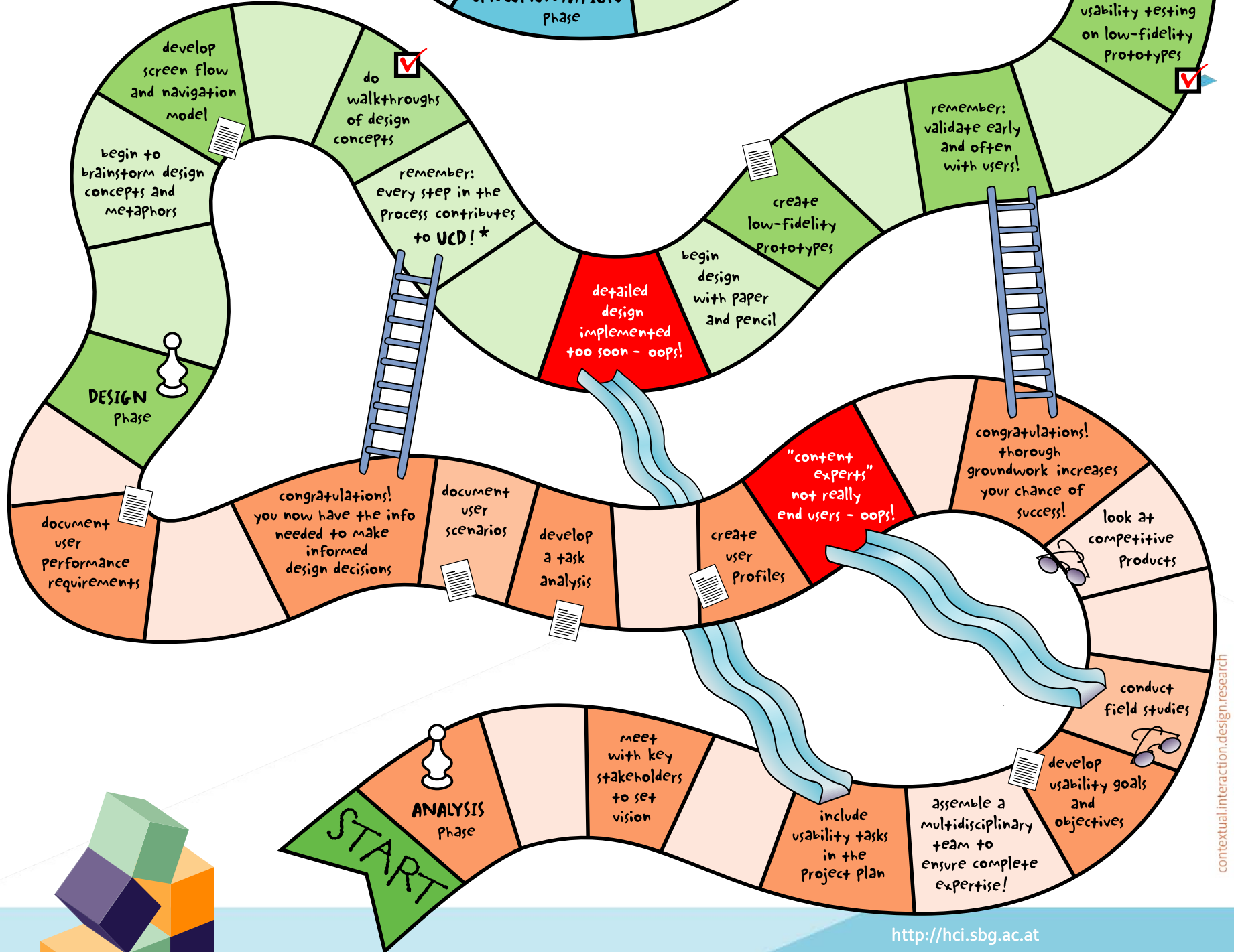
# designing the **user** experience



**upa**

usability  
professionals'  
association  
[www.upassoc.org](http://www.upassoc.org)

Disclaimer: Digital or hard copies of this file may be distributed for personal, internal corporate, or classroom use provided that copies are not made or distributed for profit or commercial advantage. Copies must bear this notice and the UPA logo. Any other use requires specific permission from UPA.  
© 2000 Usability Professionals' Association



# designing the **user** experience





# Usability Engineering Ablauf nach C. Lewis



1. Wer nutzt das System um was zu tun?
2. Wähle einen repräsentativen Task für eine "Task-centered design"
3. Plagiiere!
4. Schnelle erste Designentwürfe
5. Rethinking
6. Baue einen Prototypen/Mock-up
7. Tests mit Nutzern
8. Iteration
9. Implementierung
10. Beobachtung
11. Änderungen

# Usability Engineering Ablauf nach J. Nielsen



1. Kenne den Nutzer
  1. Individuelle Charakteristiken
  2. Aktuelle und gewünschte Tasks
  3. Funktionalanalyse
  4. Veränderung der Arbeit
2. Konkurrenzanalyse
3. Setzen von Usability Zielen
4. Paralleles Design
5. Participatory Design
6. Koordiniertes Design des gesamten Interfaces
7. Anwenden von Guidelines und Heuristiken
8. Prototyping
9. Empirisches Testen
10. Iteration (inkl. Design Rationale dokumentieren)
11. Feedback aus Feldeinsatz

# Goal Directed Design nach Cooper



## DESIGN TEAM ACTIVITIES:

- |                           |                                |                       |  |                                 |
|---------------------------|--------------------------------|-----------------------|--|---------------------------------|
| ▪ Stakeholder interviews  | ▪ Analysis                     | ▪ Refine scenarios    | ▪ High-level sketches                  | ▪ Detailed screen renderings    |
| ▪ Ethnographic user study | ▪ Personas                     | ▪ Define requirements | ▪ Define relationships of elements     | ▪ Iterative refinement          |
| ▪ Literature review       | ▪ Scenarios                    |                       | ▪ Validate with scenarios and feedback | ▪ U-test and/or user feedback   |
|                           | ▪ Other models (e.g. workflow) |                       |  | ▪ Form & Behavior Specification |

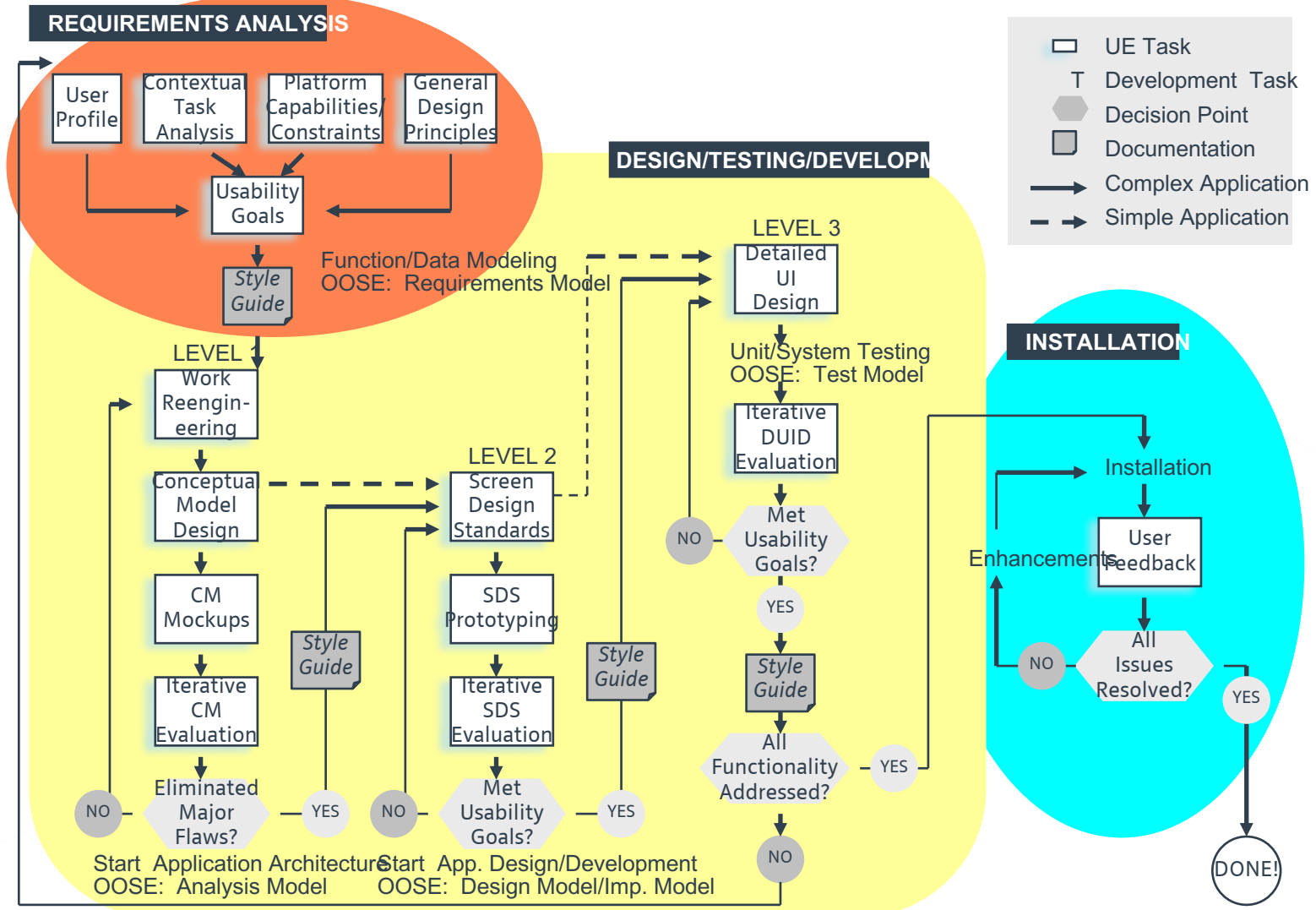
## COLLABORATION TOPICS:

- |                     |                                   |                           |   |                                |
|---------------------|-----------------------------------|---------------------------|---|--------------------------------|
| ▪ Project goals     | ▪ Persona and scenario refinement | ▪ Requirements refinement | ▪ Fitness of proposed high-level solution | ▪ Technical feasibility        |
| ▪ Research subjects |                                   |                           |   | ▪ Detailed user needs          |
|                     |                                   |                           |   | ▪ Appropriate solutions        |
|                     |                                   |                           |   | ▪ Revision based upon feedback |

# Usability Engineering nach Mayhew



## Platform



# Step-by-Step Usability Guide

## Plan

Develop a Plan

Assemble Project Team

Hold Kick-off Meeting

Write a Statement of Work

Hire a Usability Specialist

## Analyze

Evaluate Your Current Site

Learn About Your Users

Conduct Task Analysis

Develop Personas

Write Scenarios

Set Measurable Goals

## Design

Determine Site Requirements

Conduct a Content Inventory

Perform Card Sorting

Define IA

Write for the Web

Use Parallel Design

Develop a Prototype

Launch Site

## Test and Refine

Types of Evaluations

Usability Testing

Heuristic Evaluations

Implement and Retest

Create Test Plan

Prepare and Test

Data Analyses and Report

# UX Design Prozess



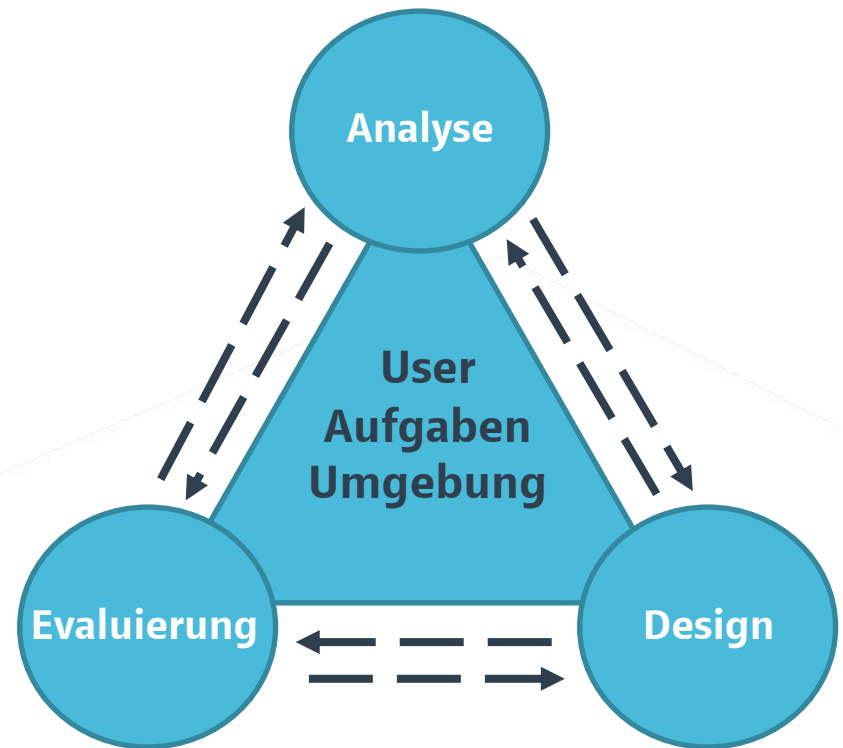
- Iterativ
  - Alle Phasen werden iteriert, um Neues zu entdecken, Probleme zu lösen, Design zu verbessern
- Evaluationszentriert
  - Ständiges evaluieren der einzelnen Schritte
- UX Lifecycle
  - Strukturiertes Rahmenmodell (Framework)
  - Vordefinierte Schritte und Aktivitäten
  - Charakterisiert die Evolution eines Interaktion Designs
  - Basierend auf Projektzielen und Ressourcen

# Was ist User-Centered Design?

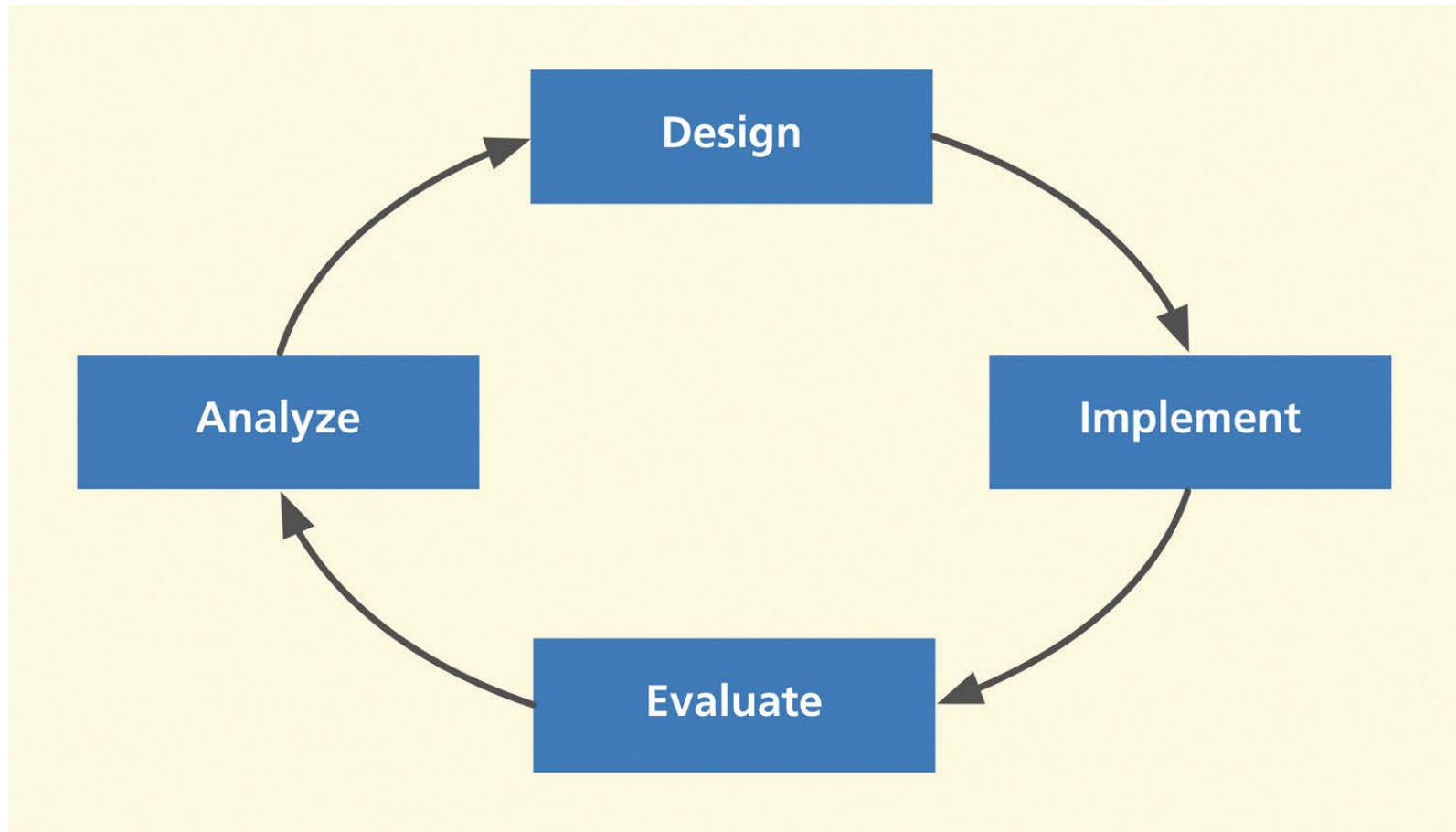


⇒ Entwicklungsansatz in  
Human-Computer Interaction

Ein interaktiver Designprozess für interaktive Produkte und Anwendungen, in dem User, ihre Aufgaben und ihre Umgebung eine zentrale Rolle spielen bei der Analyse der Anforderungen, der Entwicklung alternativer Design und bei der Evaluierung statischer oder interaktiver Prototypen.

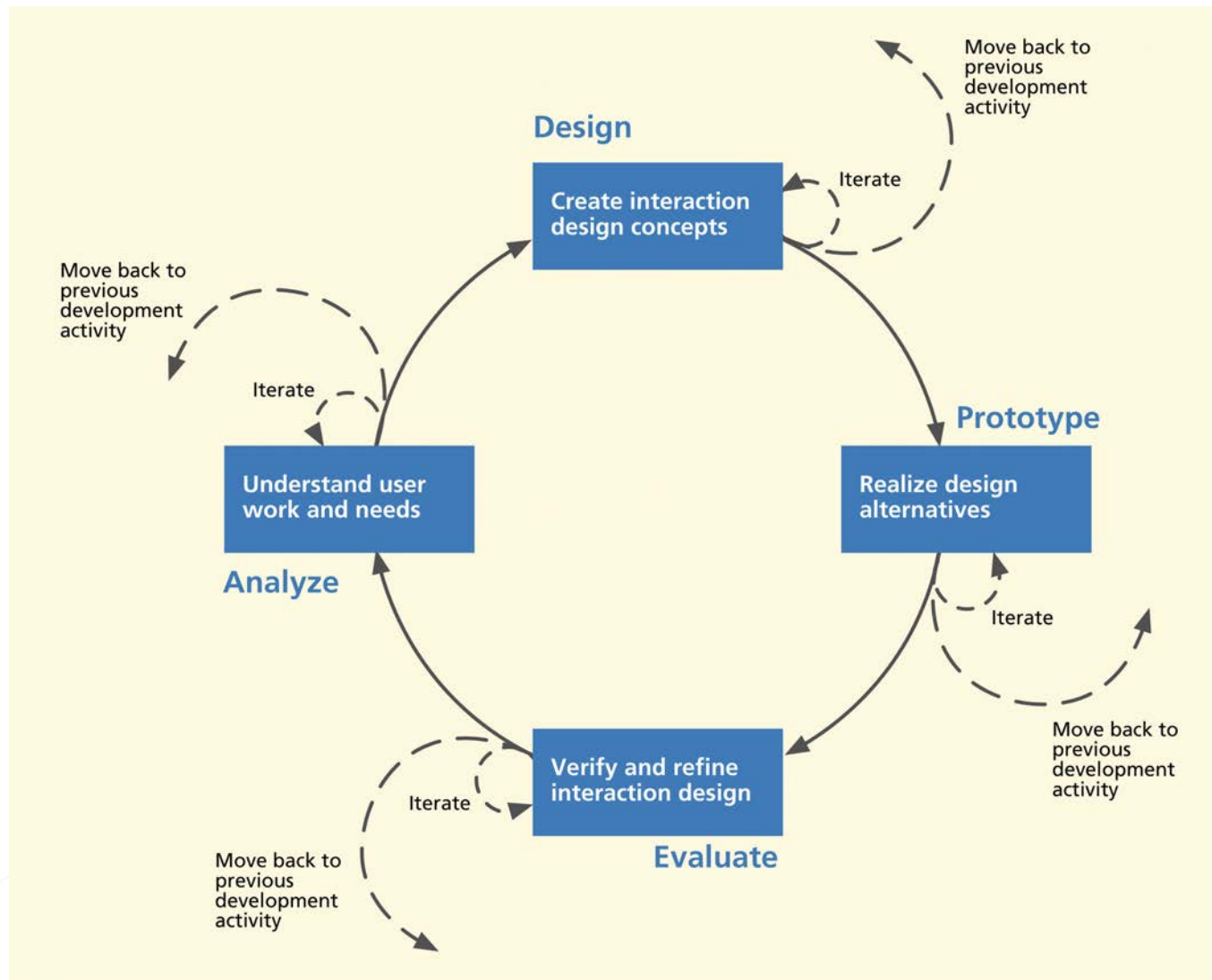


# 4 Basisaktivitäten





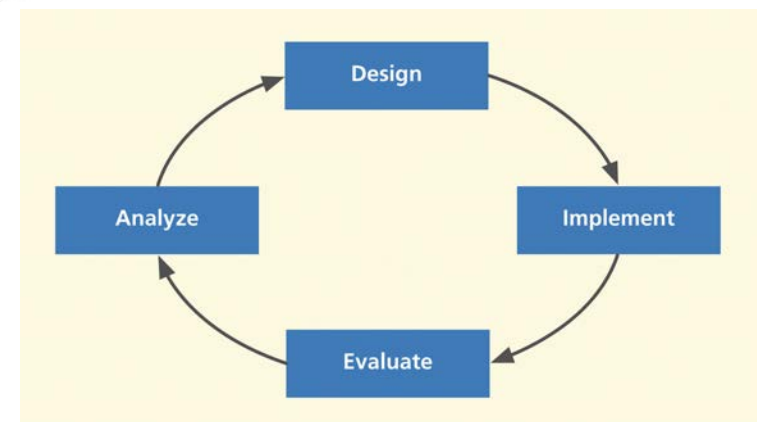
# Iterationen



# Analyse



- Verständnis über Domäne (Arbeit, Kontext, ...)
- Verständnis über Arbeit
- Verständnis über Bedürfnisse
- Methoden / Phasen
  - Contextual Inquiry
  - Contextual Analysis
  - Requirments Spezifikation
  - Synthese

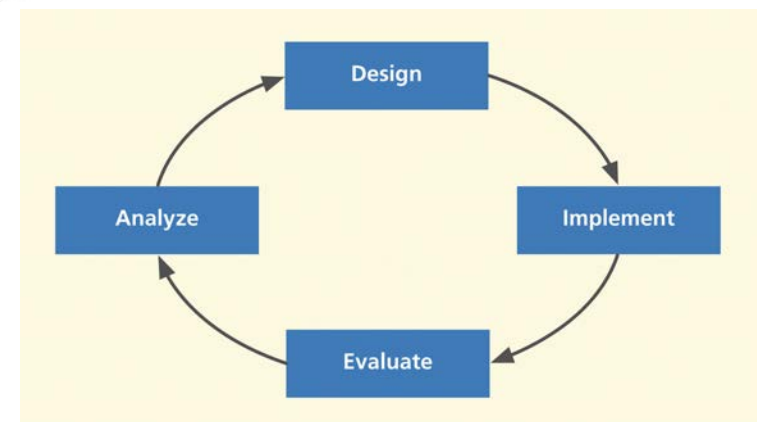


contextual.interaction.design.research

# Design



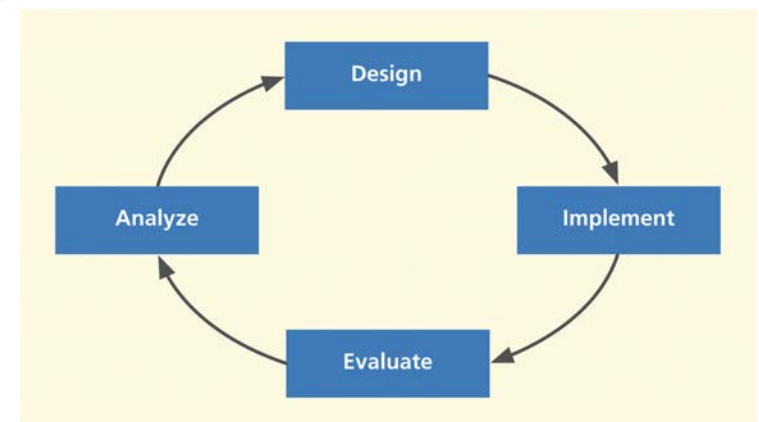
- Konzeptionelles Design
- Interaktionsverhalten
- Look & Feel
- Methoden / Phasen
  - Design Thinking, Ideation, Sketching
  - Mentale Modelle und konzeptionelles Design
  - Design Practise



# Implementation



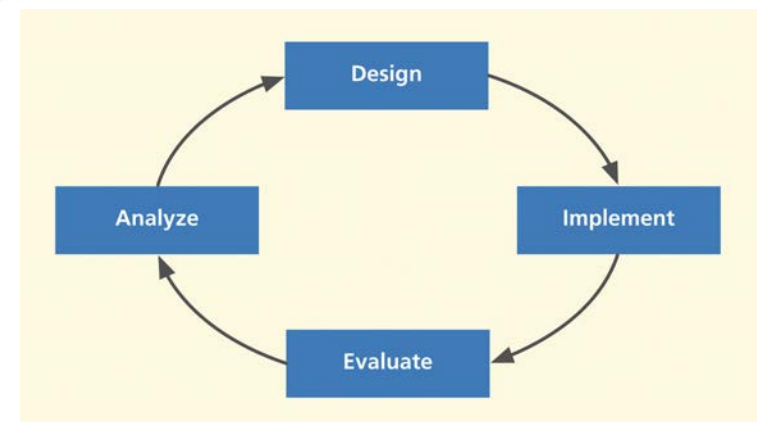
- Verschiedene Design Alternativen umsetzen
- Methoden / Phasen
  - Prototyping
  - Mock-Up
  - Experience Design



# Evaluierung



- Bewerten der Systeme
- Verifikation und Verfeinerung
- Methoden / Phasen
  - Rapid Evaluation
  - Usability / UX Testing



# Prozess



- Aktivitäten können sich überlappen
- Übergang von einer zur nächsten Aktivität muss klar strukturiert sein
- Hauptziel: Produktion voranzubringen
- Team muss entscheidungsfähig sein:
  - Wann kann/muss eine Aktivität beendet werden? (Qualitätskriterien am Ende jeder Phase)
  - Was ist der nächste Schritt / die nächste Aktivität?
  - Wann muss iteriert werden?
- Herausforderung: Limitierte Ressourcen (Zeit, Geld)

# Projekt Parameter



- Projekt Parameter beeinflussen die Wahl des Prozesses
- Risikotoleranz
  - Fehlentwicklungen
  - Fehlende Requirements
  - Nicht Adressieren von Nutzerbedürfnissen
- Je niedriger die Risikotoleranz → desto rigoroser und vollständiger muss der Prozess definiert sein
- Wichtig: Zielorientierte Prozesswahl

# Projekt Ressourcen



- Budget
  - Plan
  - Anzahl der Mitarbeiter
  - Fähigkeiten der Mitarbeiter
  - ...
- 
- Je erfahrener die Projektmitarbeiter → desto weniger streng muss der Prozess gestaltet sein

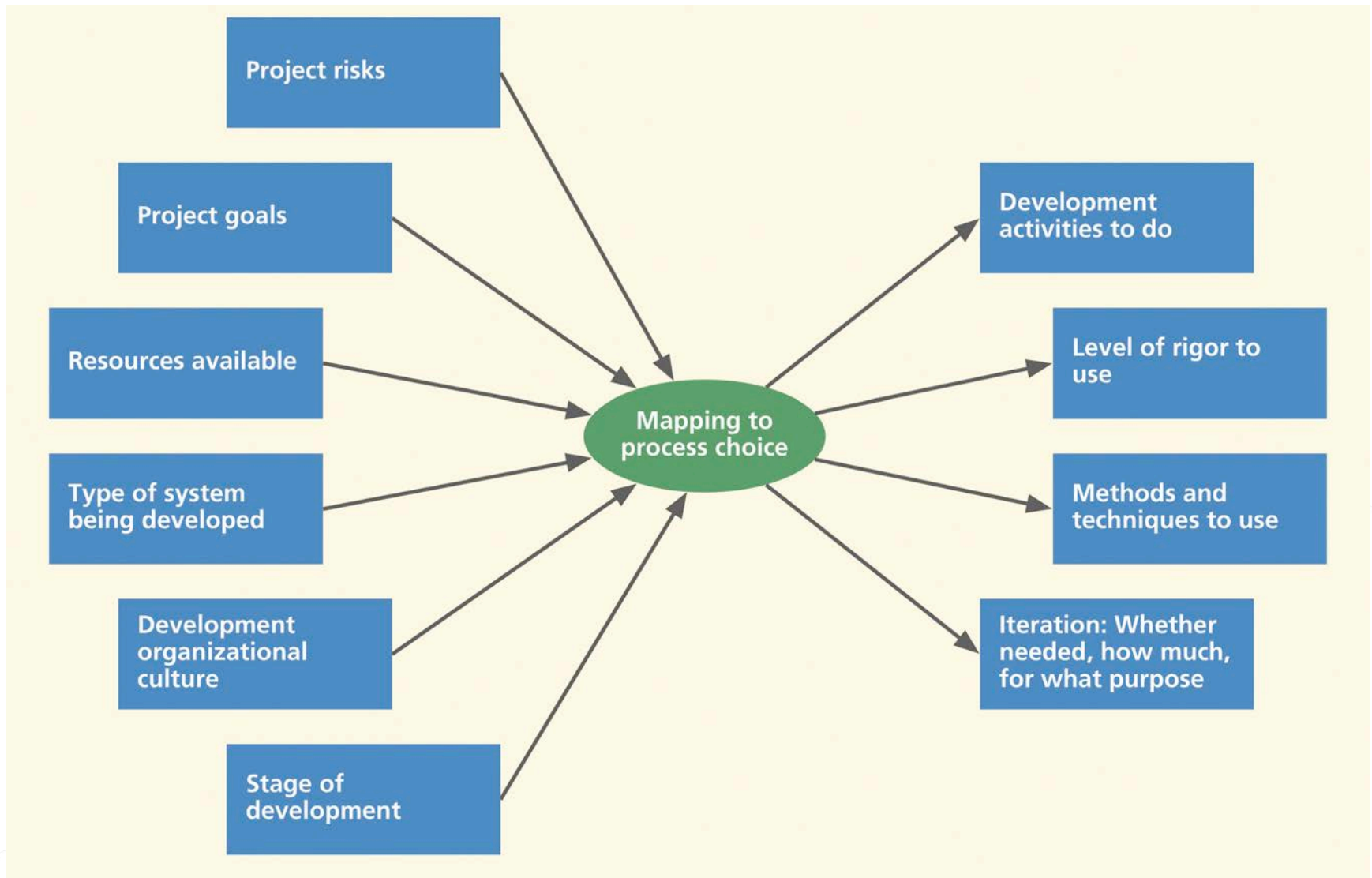


# Andere Projekt Parameter

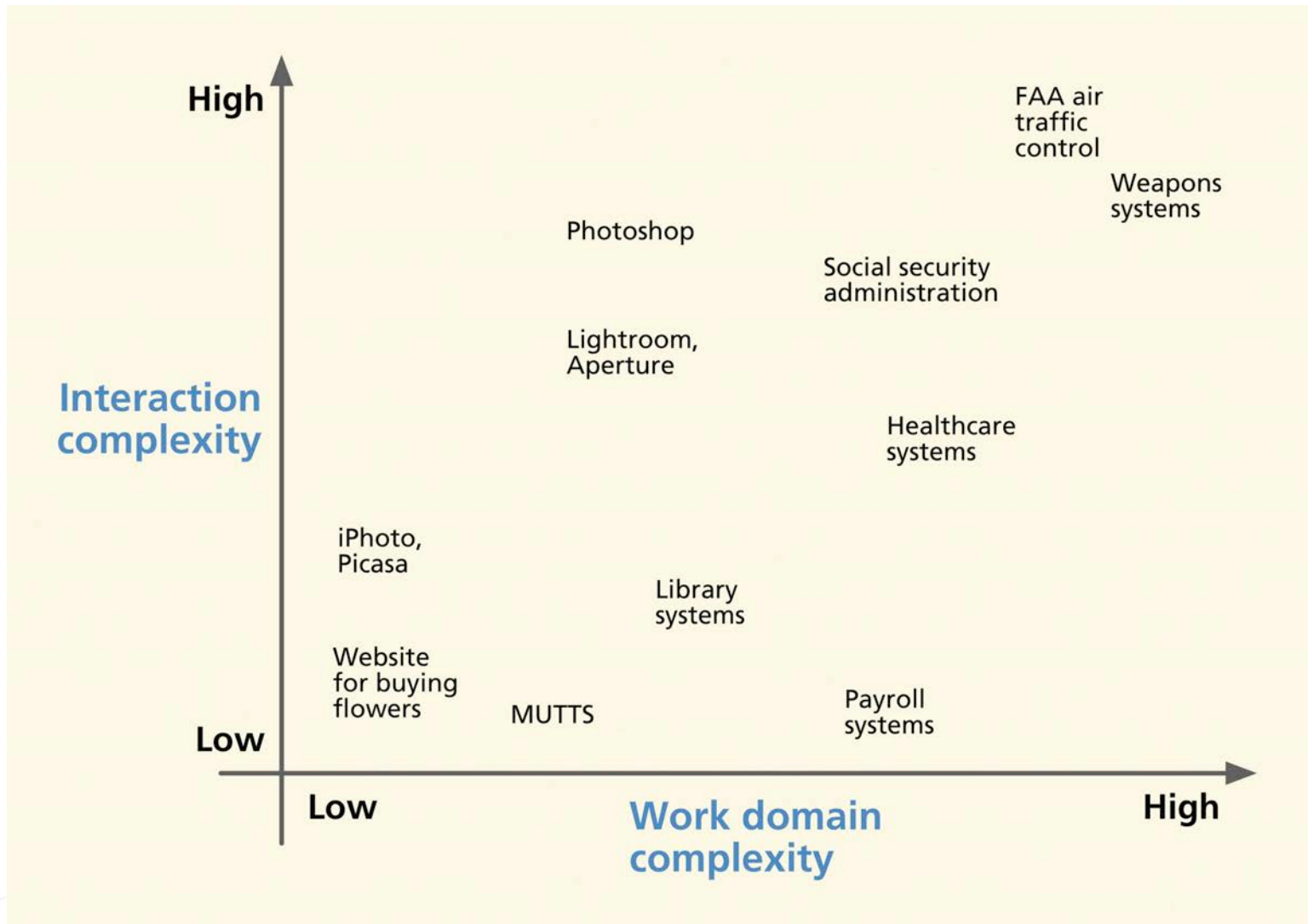


- Eigenschaften des zu gestaltenden Systems
  - Z.B. Mp3 Player vs. Flugsteuerungssystem
- Organisationsstruktur
  - Geschichte
  - Traditionen
  - Marktposition
  - Zeit bis zur Markteinführung
- Stufe des Fortschritts im Projekt

# Projekt Parameter → Prozesswahl



# Komplexitätsraum des Systems



# Interaktionskomplexität



- Komplexität der Nutzerhandlungen
- Kognitive Anstrengung
- Niedrige Interaktionskomplexität:
  - Einfache Tasks
  - Wohlbekannte Tasks
  - Immer wiederkehrende Tasks
  - z.B. Buch bei Amazon bestellen
- Hohe Interaktionskomplexität:
  - Schwierigere Tasks
  - Mehrere Tasks
  - Setzt Training oder Fähigkeiten voraus
  - z.B. Bildbearbeitung mit Adobe Photoshop

# Komplexität der Arbeitsdomäne



- Niedrige Arbeitsdomänenkomplexität:
  - Bekannte bzw. alltägliche Arbeitsdomäne
  - z.B. Internet
- Hohe Arbeitsdomänenkomplexität:
  - Komplexe Arbeitsflussmechanismen
  - Kollaborative Arbeiten
  - Abhängigkeiten und Einschränkungen
  - z.B. Halbleiterfabrik

→ Je höher die Komplexität, desto strenger der Prozess

# Komplexität der Arbeitsdomäne



## Simple Interaktion & komplexe Arbeitsdomäne:

- Fokus auf Analyse (Kontextanalyse, Modellierung, Requirements, ...)
- Benötigt Verständnis über interne Systemkomplexität
- z.B. Software für Steuererklärung

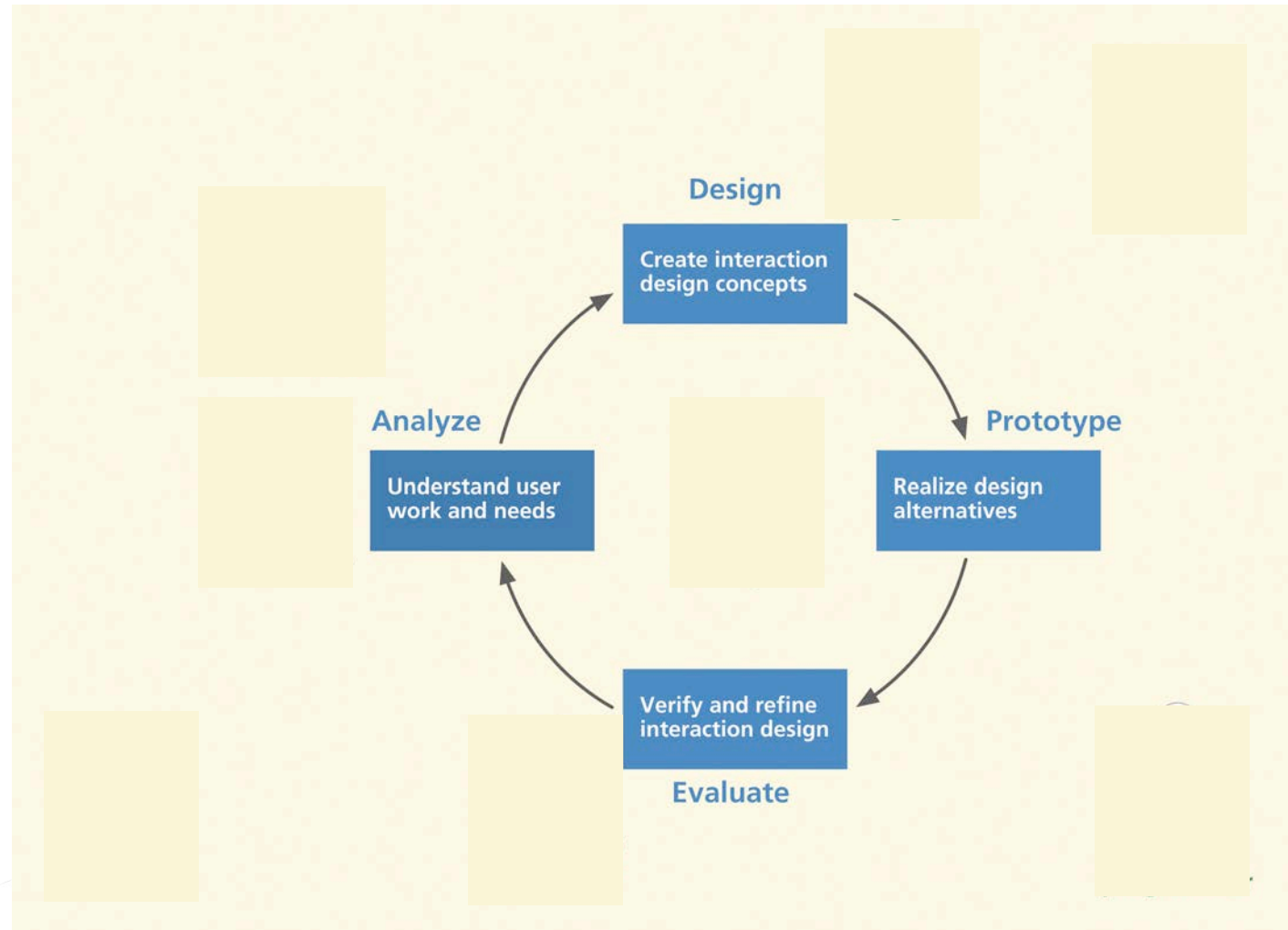
## Simple Interaktion & simple Arbeitsdomäne:

- Fokus auf emotionale Aspekte (e.g., Joy-of-use, Ästhetik, ...)
- z.B. neuer MP3 Player

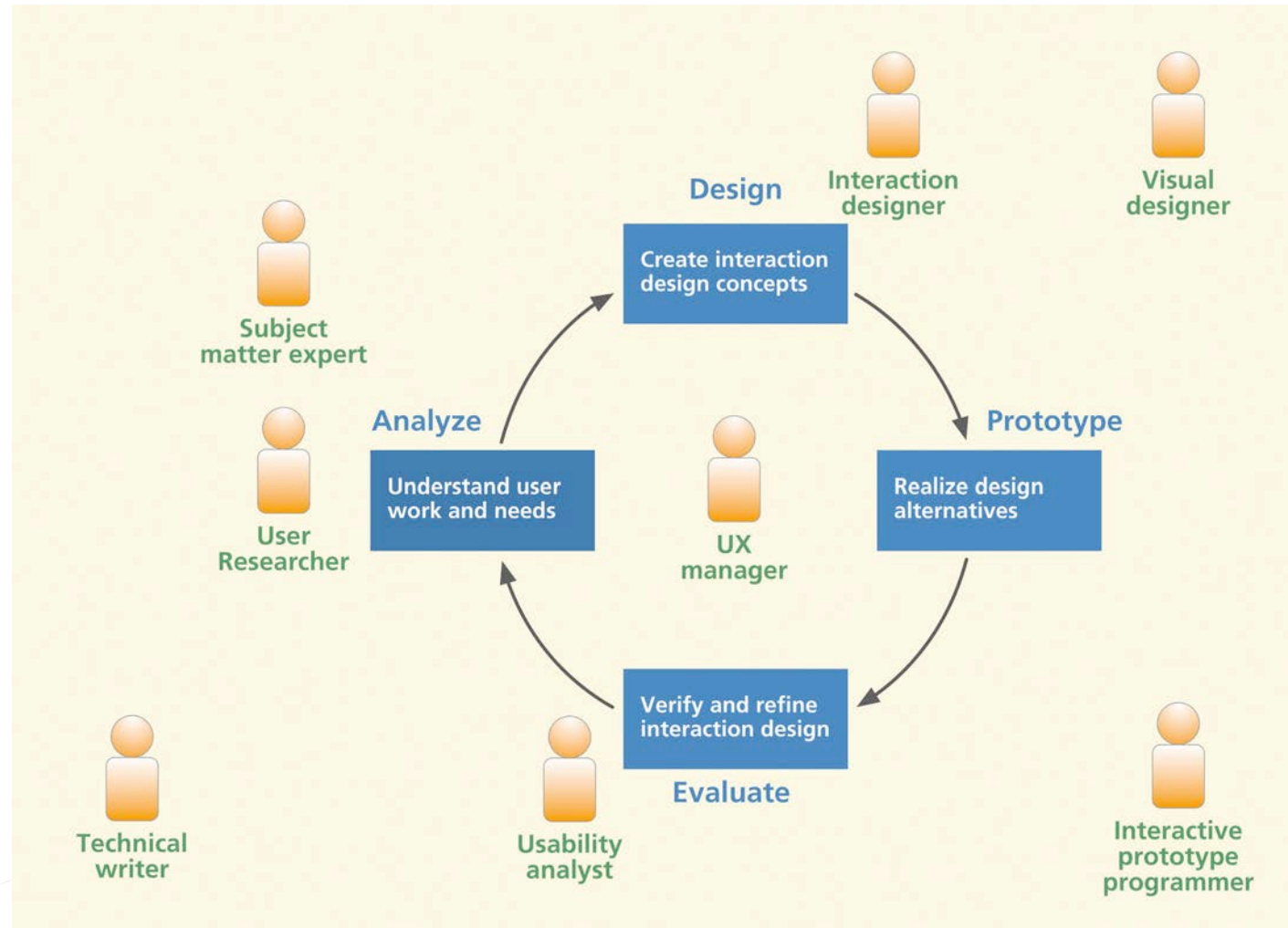
## Komplexe Interaktion & simple Arbeitsdomäne:

- Fokus auf Ideation, Design, Sketching
- Evaluation mit echten Nutzern
- Interaction Design: Task Struktur, Screen Layout, Nutzeraktionen, Metaphern
- z.B. neue Smartwatch

# User Interface Team ?



# User Interface Team



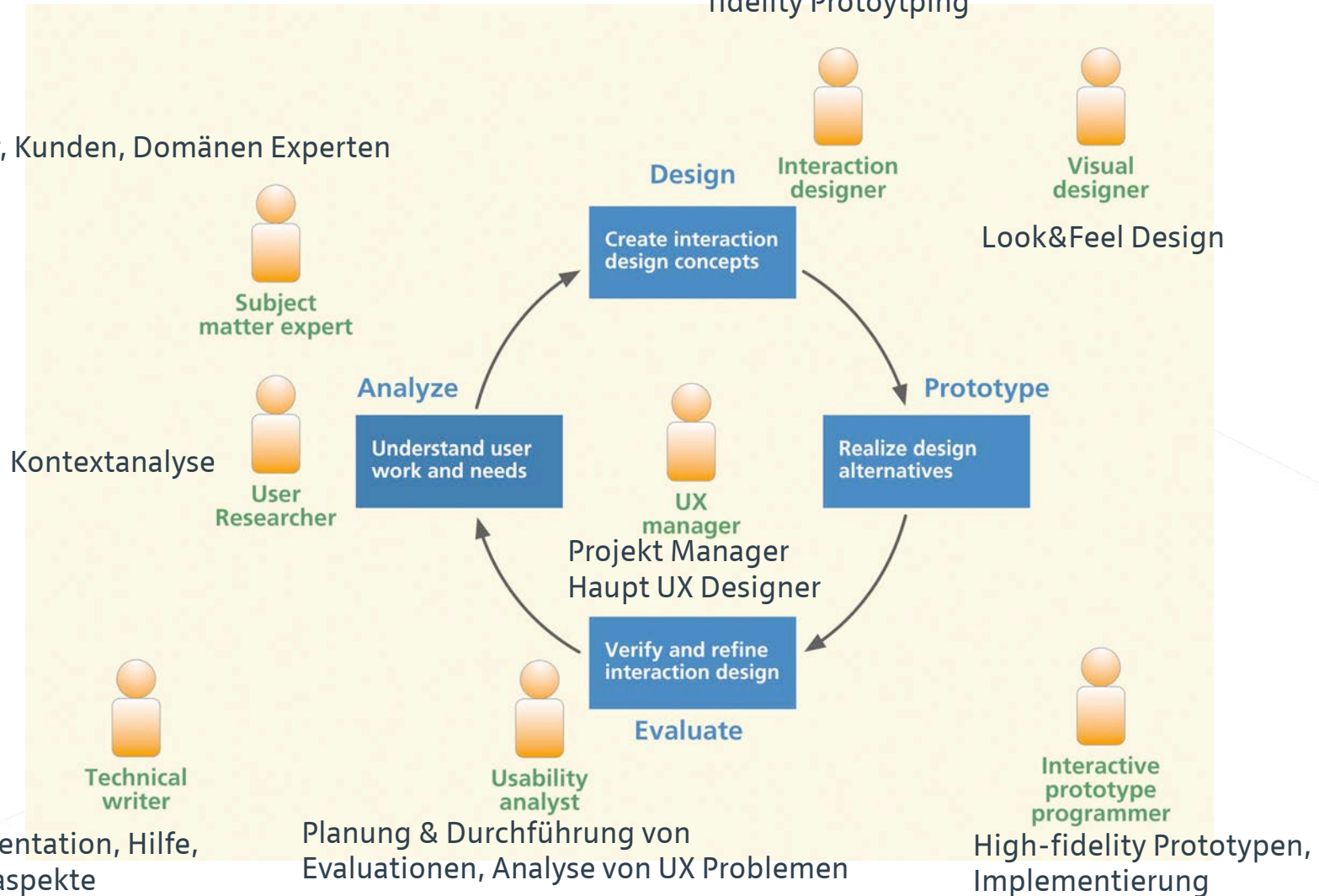


# User Interface Team

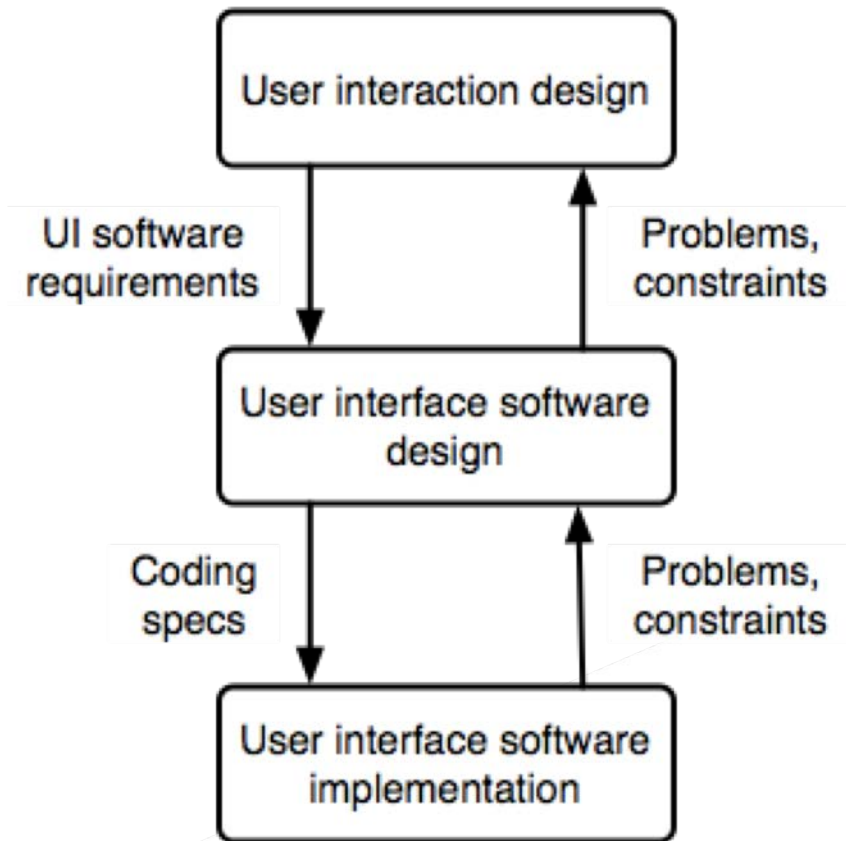


Ideation, sketching,  
konzeptionelles Design,  
detailliertes Design, low-  
fidelity Prototyping

Nutzer, Kunden, Domänen Experten



# Interaction Design ist nicht Coding

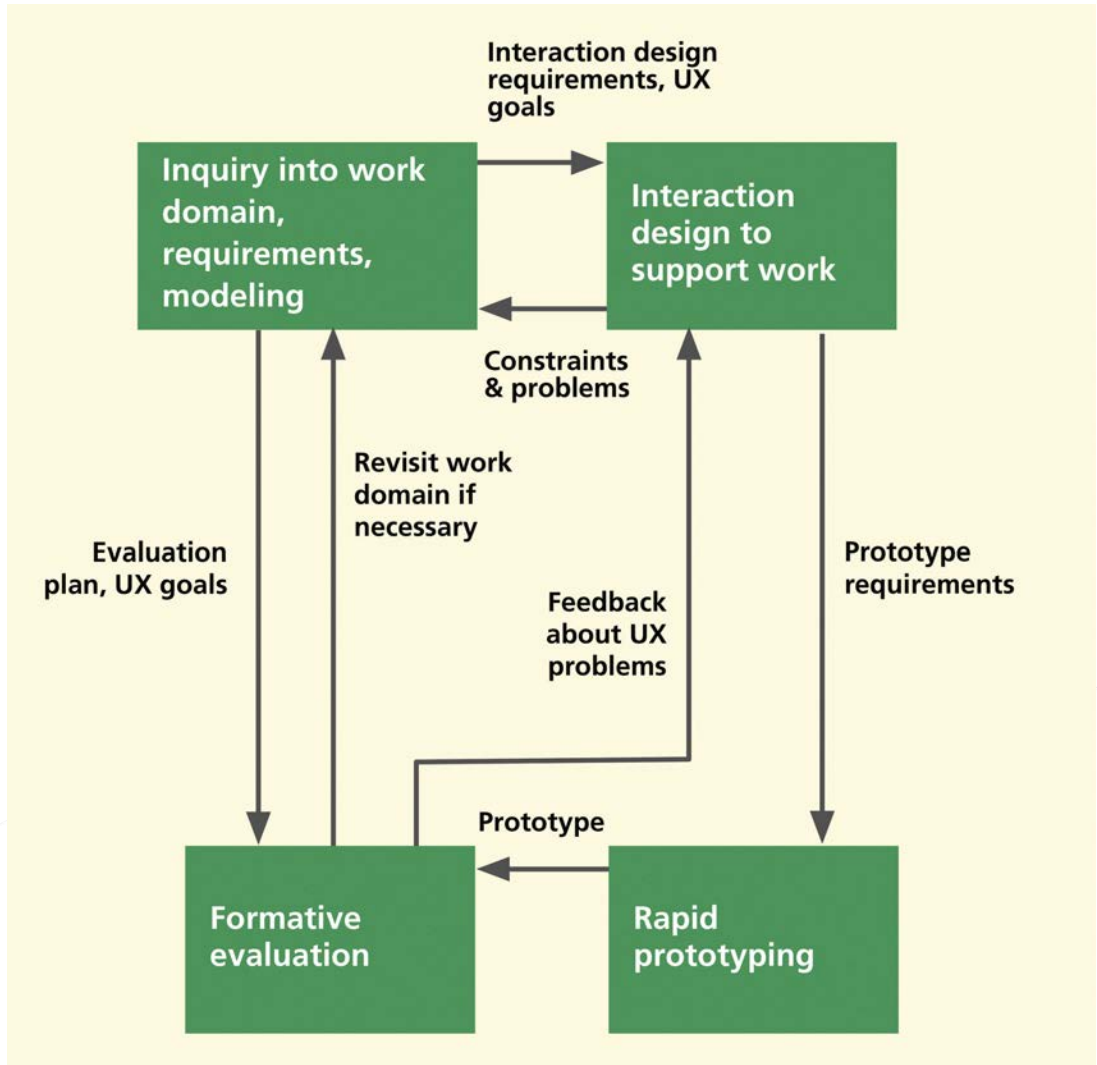


- 2 Rollen:
  - Interaction Designer
  - UI Software Designer
- Prämisse: Interaktion aus Nutzersicht sollte zu einer verbesserten UX führen als aus einer Programmierer Sicht
- Interessenskonflikt: oft nicht leicht umzusetzen!

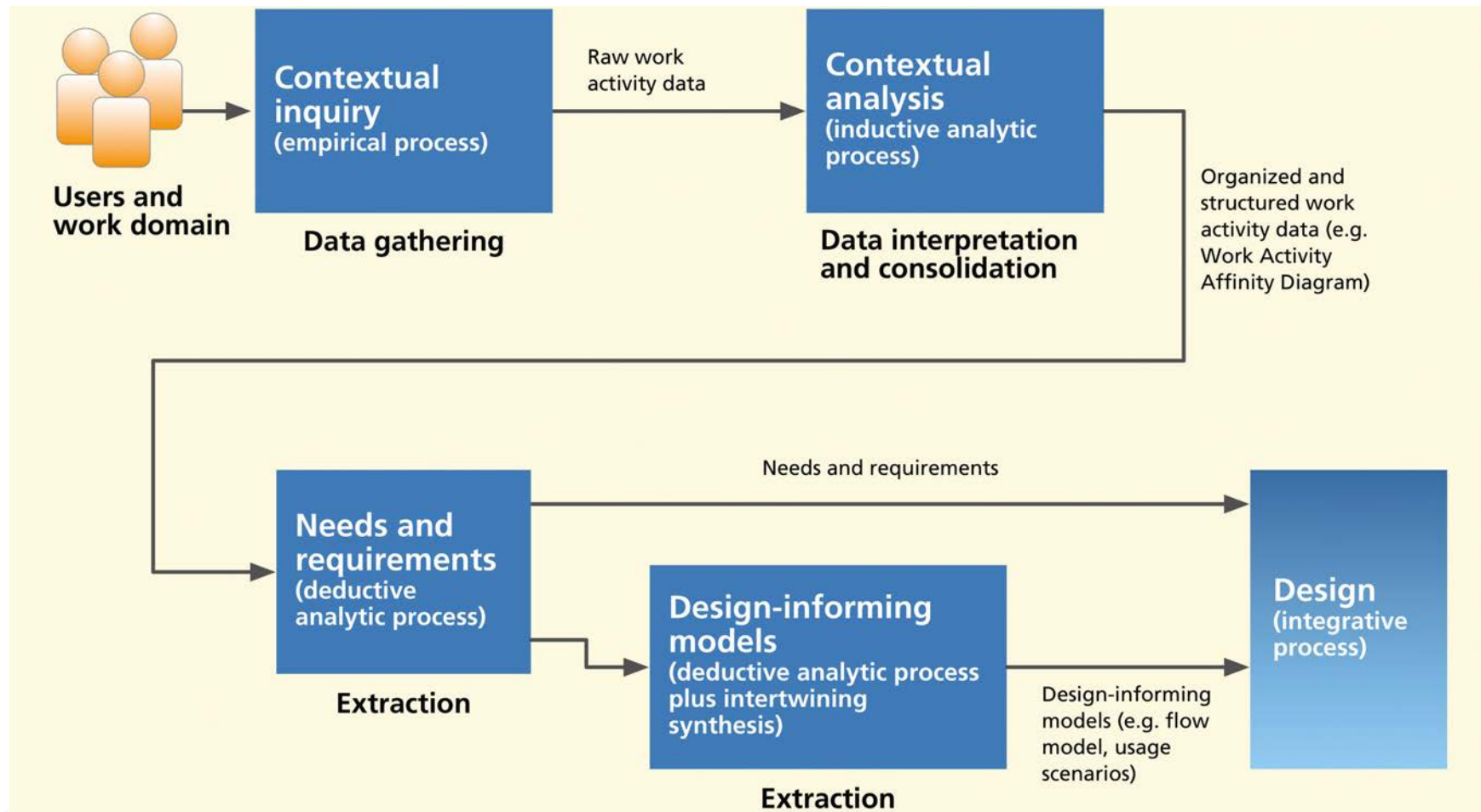
# Iteration



- UX Iteration ist NICHT mehrmals durch den Prozess zu gehen
- Iteration nur für frühe, kleine, günstige Teile des Entwicklungsprozesses



# Prozess



contextual.interaction.design.research

# Alternative: Agile Design



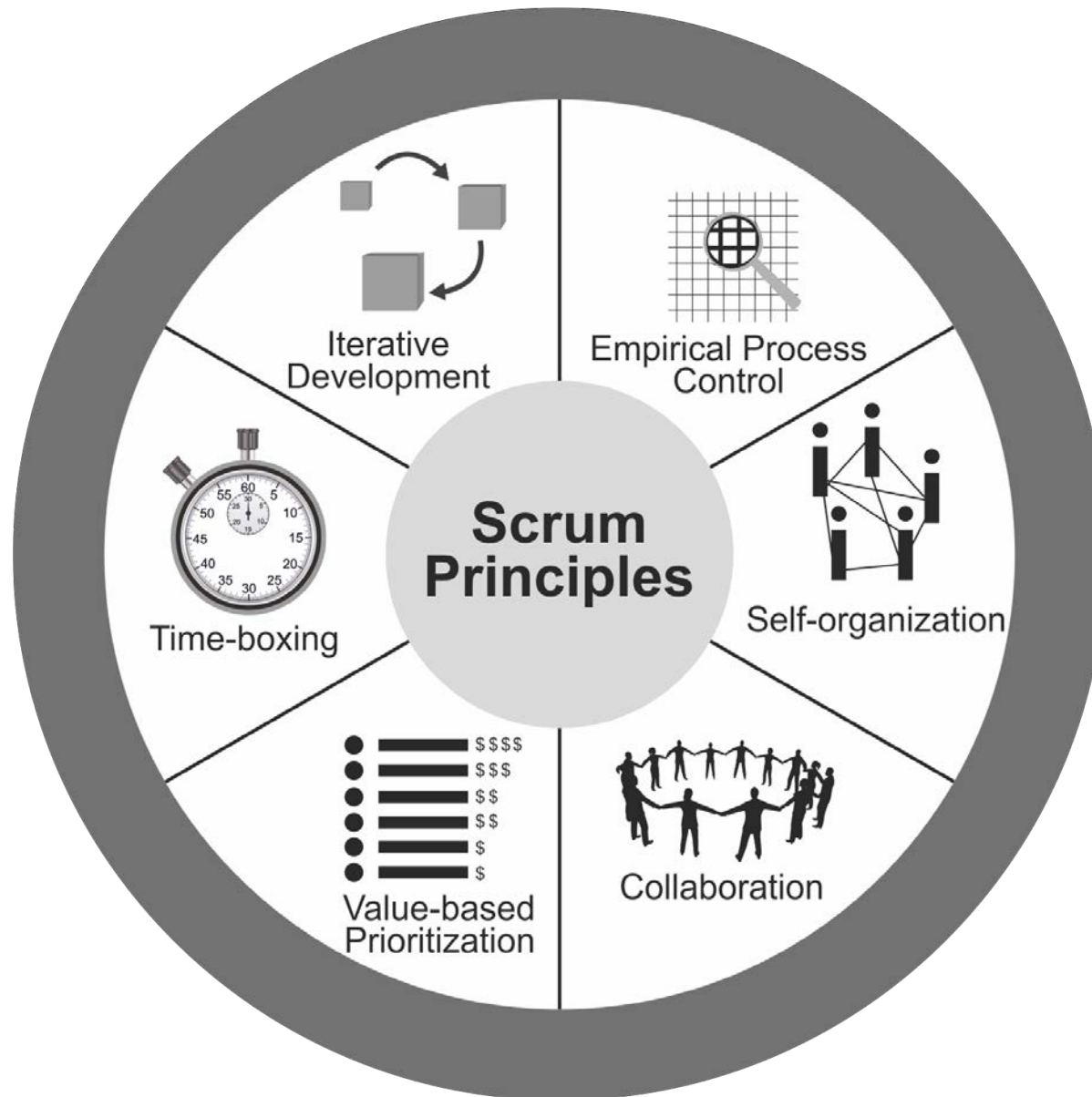
Manifest für Agile Softwareentwicklung:

1. Individuen und Interaktionen mehr als Prozesse und Werkzeuge
2. Funktionierende Software mehr als umfassende Dokumentation
3. Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
4. Reagieren auf Veränderung mehr als das Befolgen eines Plans

# Scrum



- Ein Rahmenwerk, innerhalb dessen Menschen komplexe adaptive Aufgabenstellungen angehen können, und durch das sie in die Lage versetzt werden, produktiv und kreativ Produkte mit dem höchstmöglichen Wert auszuliefern.
- Scrum Rahmenwerk besteht aus Scrum **Teams** und den mit ihnen verbundenen **Rollen, Ereignissen, Artefakten** und **Regeln**.
- Scrum basiert auf der Theorie empirischer Prozesssteuerung oder kurz "Empirie". Empirie bedeutet, dass Wissen aus Erfahrung gewonnen wird und Entscheidungen auf Basis des Bekannten getroffen werden.
- Nach: <http://www.scrumguides.org/>



# 3 Säulen



## Transparenz

- Wesentlichen Aspekte des Prozesses müssen für diejenigen sichtbar sein, die für das Ergebnis verantwortlich sind.
- Gemeinsamen Standard, damit die Betrachter ein gemeinsames Verständnis teilen.
- Gemeinsame Prozesssprache
- Gemeinsames Verständnis der Definition von "Done"

## Überprüfung

- Scrum Anwender müssen die Scrum Artefakte und den Fortschritt ständig in Bezug auf die Erreichung des Sprint-Ziels überprüfen, um unerwünschte Abweichungen zu erkennen
- Keine häufige Überprüfung
- Gewissenhaft durch fähige Prüfer dort vorgenommen wo die Arbeit verrichtet wird.

## Anpassung

- Bei Abweichung in Überprüfung → Anpassung
- So schnell wie möglich



# Ereignisse



4 formale Ereignisse für Überprüfung und Anpassung:

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospektive

# Scrum Team



- Besteht aus **Product Owner, Entwicklungsteam, Scrum Master**
- Scrum Teams sind **selbstorganisierend** und **interdisziplinär**
- Teams entscheiden selbst, wie sie ihre Arbeit am besten erledigen
- Teams verfügen über alle Kompetenzen, die erforderlich sind, um die Arbeit zu erledigen ohne Abhängigkeiten von aussen
- Flexibilität, Kreativität und Produktivität optimieren

# Product Owner



- Für Wertmaximierung und Arbeit des Entwicklungsteams verantwortlich
- Für das Management des Product Backlogs verantwortlich:
  - Product Backlog Einträge klar zu formulieren
  - Die Einträge im Product Backlog so zu sortieren, dass Ziele und Missionen optimal erreicht werden können
  - Den Wert der Arbeit zu optimieren, die das Entwicklungsteam erledigt;
  - Das Sicherstellen, dass das Product Backlog sichtbar, transparent und für alle klar ist sowie zeigt, woran das Scrum Team als nächstes arbeiten wird
  - sicherzustellen, dass das Entwicklungsteam die Product Backlog Einträge im erforderlichen Maße versteht
- Einzelperson
- Verantwortlich
- Entscheidungsträger

# Entwicklungsteam



- Profis, die am Ende eines jeden Sprints ein fertiges Inkrement übergeben
- Selbstorganisierend
- Interdisziplinär
- Alle Mitglieder sind "Entwickler" egal was sie machen
- Es gibt keine weiteren Unterteilungen
- Als Team rechenschaftspflichtig
- Min. 3 Entwickler, max. 9 Entwickler

# Scrum Master



- Für Verständnis und die Durchführung von Scrum verantwortlich
- Sorgt dafür, dass das Scrum Team die Theorie, Praktiken und Regeln von Scrum einhält
- Hilft dabei die Zusammenarbeit so zu optimieren

# Scrum Ereignisse



- Alle Ereignisse haben eine zeitliche Beschränkung (**Time Box**)
- Dauer eines Sprints steht zu seinem Beginn fest und darf weder gekürzt noch verlängert werden
- Die anderen Ereignisse dürfen beendet werden, sobald sie ihren Zweck erfüllt haben
- Ereignisse sind dazu gedacht, an den kritischen Stellen Transparenz und Überprüfung zu ermöglichen

# Sprint



- Das Herz von Scrum ist der **Sprint**, eine Time Box von **maximal einem Monat**, innerhalb dessen ein fertiges ("Done"), nutzbares und potenziell auslieferbares Produkt-Inkrement hergestellt wird.
- Alle Sprints innerhalb eines Entwicklungsvorhabens sollten die gleiche Dauer haben
- Der neue Sprint startet sofort nach dem Abschluss des vorigen Sprints
- Ein Sprint beinhaltet und umfasst das **Sprint Planning**, die **Daily Scrums**, die **Entwicklungsarbeit**, das **Sprint Review** und die **Sprint Retrospektive**.
- Sprint kann durch Project Owner abgebrochen werden

# Sprint Planning



- Sprint Planning: maximal 8 Stunden
- Was ist in dem Produkt-Inkrement des kommenden Sprints enthalten?
- Wie wird die für die Lieferung des Produkt-Inkrementes erforderliche Arbeit erreicht?
- Eingangsgrößen für das Meeting:
  - Product Backlog
  - das neueste Produkt-Inkrement
  - Kapazität des Entwicklungsteams im Sprint
- Ausformulieren des Sprint Ziels durch das gesamte Team



# Daily Scrum



- Time Box von 15 Minuten, innerhalb derer das Entwicklungsteam seine Aktivitäten synchronisiert und an der Planung für die nächsten 24 Stunden arbeitet
- An jedem Tag zur gleichen Uhrzeit am gleichen Ort
  - Was habe ich gestern erreicht , das dem Entwicklungsteam hilft, das Sprint-Ziel zu erreichen?
  - Was werde ich heute erledigen, um dem Entwicklungsteam bei der Erreichung des Sprint-Ziels zu helfen?
  - Sehe ich irgendwelche Hindernisse [Impediments], die mich oder das Entwicklungsteam vom Erreichen des Ziels abhalten?

# Sprint Review



- Am Ende eines Sprints wird ein Sprint Review abgehalten, um das [Produkt-]Inkrement zu überprüfen und das Product Backlog bei Bedarf anzupassen
- Für einen einmonatigen Sprint wird für dieses Meeting eine Time Box von 4 Stunden
- Product Owner erklärt, welche Product Backlog-Einträge "Done" sind
- Product Owner stellt den aktuellen Stand des Product Backlogs dar
- Ergebnis des Sprint Reviews ist ein überarbeitetes Product Backlog

# Sprint Retrospektive



- Bietet dem Scrum Team die Gelegenheit, sich selbst zu überprüfen und einen Verbesserungsplan für den kommenden Sprint zu erstellen
- Findet zwischen dem Sprint Review und dem nächsten Sprint Planning statt
- Time Box von drei Stunden

# Scrum Artefakte



- Product Backlog
- Sprint Backlog
- Inkrement

# Product Backlog



- Im Product Backlog werden alle Features, Funktionalitäten, Verbesserungen und Fehlerbehebungen aufgelistet, die die Änderungen an dem Produkt in zukünftigen Releases ausmachen.
- Geordnete Liste von allem, was in dem Produkt enthalten sein kann
- Einzige Anforderungsquelle für alle Änderungen am Produkt
- Product Backlog ist niemals vollständig
- Verfeinerung [Refinement]: Details zu Einträgen hinzugefügt, Schätzungen erstellt, oder die Reihenfolge der Einträge bestimmt

# Sprint Backlog



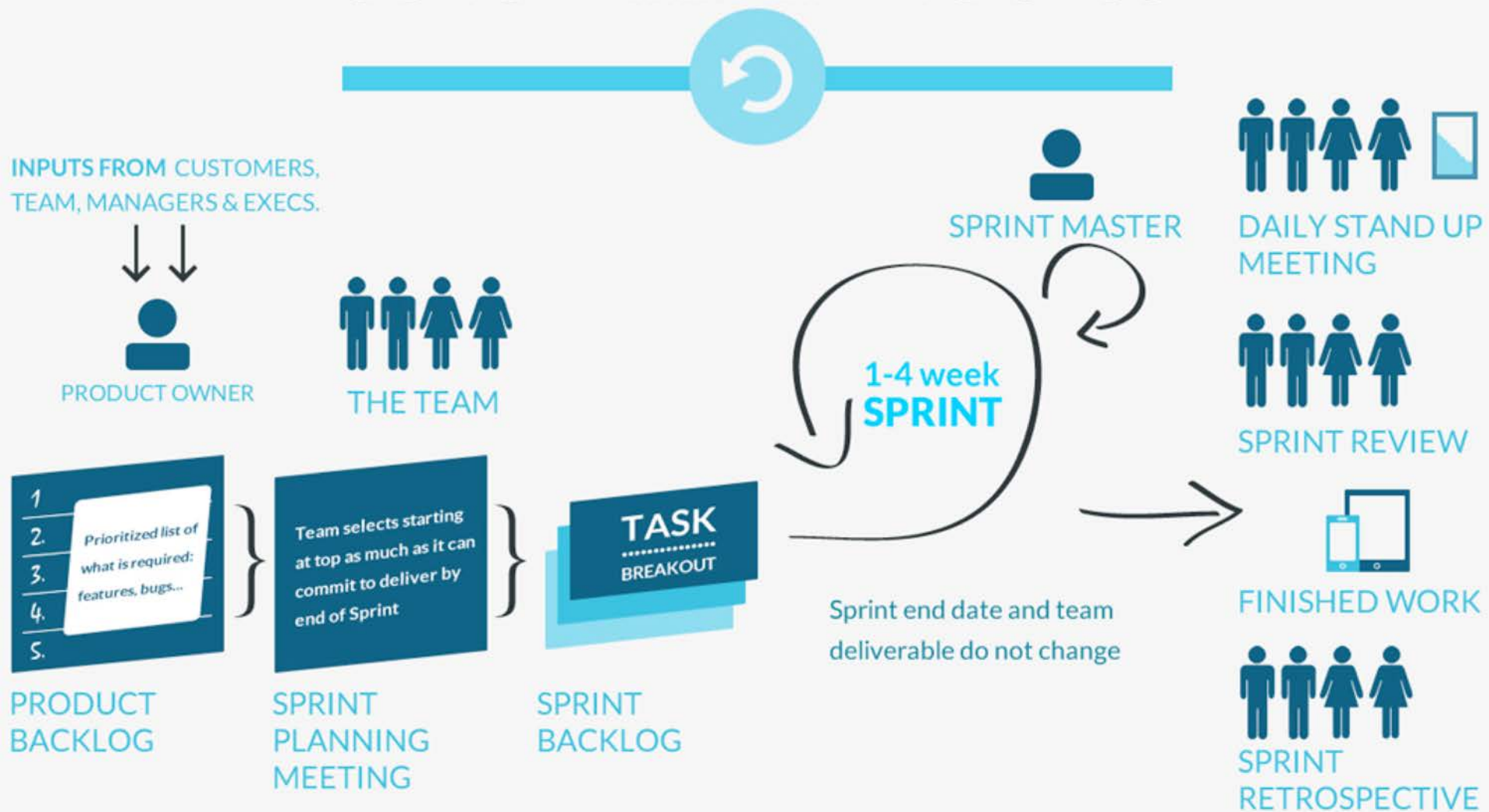
- Das Sprint Backlog ist die Menge der für den Sprint ausgewählten Product Backlog-Einträge, ergänzt um den Plan für die Lieferung des Produkt-Inkrementes sowie zur Erfüllung des Sprint-Ziels.
- Das Sprint Backlog ist eine Prognose des Entwicklungsteams darüber, welche Funktionalität im nächsten Inkrement enthalten sein wird, sowie über die erforderliche Arbeit, um diese Funktionalität in einem fertigen „Done“ Inkrement zu liefern.

# Inkrement



- Das Inkrement ist das Ergebnis aus allen in einem Sprint fertiggestellten Product Backlog-Einträgen und dem Resultat der Inkremente aller früheren Sprints.
- Am Ende eines Sprints muss das neue Inkrement "Done" sein

# the SCRUM SOFTWARE DEVELOPMENT PROCESS





# Design Prinzipien im Prozess



# Typisches Problem



- Viele *Stakeholder*
- ... mit unterschiedlichen Vorstellungen und Prioritäten.
- Die Anforderungen sind komplex und miteinander verwoben.
- Das Problem ist schwer einzukreisen und verändert sich.
- Das Problem ist "einmalig".
- Es gibt im Voraus keinen Hinweis, was eine optimale Lösung ist.

# Lösungsansatz



- Definiere eine gemeinsame Vision.
- Dokumentiere Ideen und kommuniziere sie.
- Binde Stakeholder ein.
- Mache kleine Schritte, überprüfe dein Ergebnis und iteriere sie.

# User-centred Design Prinzipien



1. Die Interessen der Nutzer vertreten.
2. Interdisziplinäre Zusammenarbeit fördern.
3. Das Problem verstehen und eingrenzen.
4. Eine gemeinsame Lösungsidee entwickeln und kommunizieren.
5. Verschiedene Lösungen aufzeigen und greifbar machen.
6. Ideen überprüfen, verfeinern, verwerfen.
7. Eine (neue) angemessene Lösung finden.

# Noch Fragen...





# Kontakt



Dr. Alexander Meschtscherjakov

[alexander.meschtscherjakov@sbg.ac.at](mailto:alexander.meschtscherjakov@sbg.ac.at)

Center for HCI

Department computer Sciences

University of Salzburg

Sigmund-Haffner-Gasse 18

5020 Salzburg

Österreich