

# Human Centered Requirements Engineering – Disziplinübergreifend Synergien organisieren

Hans-Georg Hopf

[hans-georg.hopf@th-nuernberg.de](mailto:hans-georg.hopf@th-nuernberg.de)

Technische Hochschule Nürnberg Georg Simon Ohm  
Nürnberg

## ZUSAMMENFASSUNG

Anforderungen sind die Basis für jede Produktentwicklung. Anforderungen geben der Produktentwicklung Ziel und Richtung. Die Qualität eines Produktes wird erkennbar durch den Vergleich mit den ursprünglichen Anforderungen: *Quality is conformance with requirements*. Der Erhebung der Anforderungen, der Anforderungsermittlung, kommt dementsprechend eine hohe Bedeutung in der Produktentwicklung zu. Eigenschaften des Produktes umfassen funktionale Aspekte und nicht-funktionale Eigenschaften. Funktionale Produkteigenschaften beschreiben, was einem Benutzer mit dem Produkt möglich gemacht werden soll, welcher Funktionsumfang das Produkt einem Benutzer zur Verfügung stellt. Viele Methoden und Techniken im Softwareengineering sind auf das Erkennen und Spezifizieren der Produktfunktionalität gerichtet. Zu den nicht-funktionalen Eigenschaften gehören Qualitätseigenschaften. Es gibt eine Vielzahl von Qualitätseigenschaften, die für ein Produkt festgelegt werden können. Man kann diese Qualitätseigenschaften in unterschiedliche Klassen einteilen: Eigenschaften, die sich auf das Produkt selbst beziehen und die Produktqualität betreffen (*Product Quality*) und Eigenschaften, die die Qualität in der Produktnutzung zum Thema haben (*Quality in Use*). Gerade diese Anforderungen, die sich auf die Produktnutzung, also auf Usability und User Experience beziehen, sind häufig für den Produkterfolg von größter Bedeutung.

Insofern sucht man Möglichkeiten, das klassische Requirements Engineering zu erweitern und Usability Engineering bzw. User Experience Methoden aber auch kreative Ideenfindungsprozesse und Innovationsprozesse zu integrieren. Mit der menschenzentrierten Gestaltung wird schon im Entwicklungsprozess einer hohen Qualität in der Nutzung in besonderer Weise Rechnung getragen. Erhobene Nutzungsanforderungen werden in einem iterativen Prozess evaluiert. Damit wird das Entwicklungsergebnis qualitätsgesichert. Aber auch andere Methoden rücken in das Blickfeld. So wird in letzter Zeit auch Design Thinking als Möglichkeit gesehen, besonders Innovation und Ideenfindung in den Entwicklungsprozess speziell natürlich in den Prozess der Anforderungsanalyse zu integrieren.

Mit dem hier vorgestellten Human Centered Requirements Engineering soll ein Weg zur nahtlosen Integration verschiedener

Vorgehensweisen aus Usability Engineering, Softwareengineering und Innovationsmanagement aufgezeigt werden. Durch Verzahnung spezifischer Vorgehensmodelle sollen disziplinübergreifende Synergien organisiert werden. Dieser Erkenntnis folgend soll im Folgenden eine Modell-Integration vorgestellt werden, die sowohl Softwareingenieure als auch Designer abholt und sie in die Lage versetzt, danach in „synchronisierten mentalen Modellen“ koordiniert in einem Entwicklungsprojekt zusammenzuarbeiten.

## KEYWORDS

requirements engineering, human centered design, usage centered design, design thinking

## 1 AUSGANGSSITUATION

Anforderungen sind die Basis für jede Produktentwicklung. Anforderungen geben der Produktentwicklung Ziel und Richtung. Die Qualität eines Produktes wird erkennbar durch den Vergleich mit den ursprünglichen Anforderungen: *„Quality is conformance with requirements“*. Der Erhebung der Anforderungen, der Anforderungsermittlung, kommt dementsprechend eine hohe Bedeutung in der Produktentwicklung zu.

Requirements Engineering ist eine im IT-Bereich schon lange methodisch hinterlegte und etablierte Ingenieurdisziplin. Da der Projekterfolg von einer guten Anforderungsermittlung und -analyse ganz wesentlich abhängt, ist es im Softwareengineering eine mit sehr viel Expertise und Sorgfalt durchgeführte Aktivität, unabhängig vom gewählten Vorgehensmodell (klassische Vorgehensweise, agiles Vorgehen).

Mit der zunehmenden Bedeutung der Usability für den wirtschaftliche Erfolg eines Produktes wurde die Zusammenarbeit mit Usability Experten und Interaktionsdesignern immer wichtiger. Zwischen den Disziplinen gab und gibt es aber im Zusammenwirken immer wieder Probleme. Oft verlaufen die Usability Engineeringprozesse und die Softwareengineeringprozesse ohne ausreichende Kommunikation und Koordination. Häufig wird der Mangel an Koordination von Entwicklungsaktivitäten, Synchronisation von Projektplanungen, Kommunikation zwischen den beteiligten UE / SE – Entwicklern, Abstimmung bzgl. Abhängigkeiten und Randbedingungen, Austausch bzgl. Änderungen und der Mangel bzgl. Austausch / Verständnis von Design- und Entwicklungsmethoden beklagt.

Die Integration von Usability Engineering in die Softwareentwicklung ist ein schon lange bearbeitetes Thema. 2005 ist mit dem Buch „Human-Centered Software Engineering“ herausgegeben von Ahmed Seffah et. al. (siehe [6]) eine umfassende Zusammenstellung von vielen verschiedenen Integrationsansätzen veröffentlicht worden. Pradha S. Pyla et al. stellen hier z.B. ein „Event Driven Design

Representation Framework for integration of Usability and Software Engineering Life cycle“ vor (siehe [3]). Der Ansatz zur Integration von Usability Engineering und Software Engineering war nicht darauf ausgelegt, die beiden Ansätze zu einem gemeinsamen Prozess zu verschmelzen. Es wurde in diesem Integrationsvorschlag auf die Koordination von Aktivitäten, Zeitvorgaben, Lösungsraum und Zielsetzungen Wert gelegt und ein „shared design representation and management“ für die beiden einzelnen Disziplinen etabliert. Damit sollte den oben beschriebenen Mängeln begegnet werden. In diesem Ansatz gab es aber keine detaillierten Vorschläge für das konkrete methodische Umsetzen der Koordination.

In den letzten Jahren ist das Thema Design Thinking in den Blick von Designern aber auch Softwareentwicklern geraten. Jennifer Hehn et. al. untersuchten 2020 die Möglichkeiten und Notwendigkeiten Design Thinking in das Requirements Engineering zu integrieren (siehe [2]). Es wird festgestellt: „Ein Grundproblem des Requirements Engineerings ist es die unscharfen Bedürfnisse und unbestimmten, oft sich verändernden Anforderungen der verschiedenen Stakeholder zu entdecken“. Da Design Thinking ursprünglich eine menschenzentrierte rapid-prototyping Methode für innovatives Design ist, wird in dieser Methode ein vielversprechender Ansatz gesehen dieses Problem zu lösen. Für die Integration werden drei Modelle entwickelt: Upfront DT: Methode wird in der Anfangsphase eines Projekts genutzt, um unklare Benutzerbedürfnisse zu klären und eine Produktversion zu entwickeln. Infused DT: Methode wird in einem existierenden RE-Prozess genutzt, um neue Einsichten zu gewinnen oder unscharfe Anforderungen zu präzisieren. Continuous DT: Methode beinhaltet den kontinuierlichen Einsatz von DT und RE Elementen, um eine Gesamtübersicht von Benutzerbedürfnissen über Lösungsvisionen bis zur funktionalen Lösung zu gewinnen. Ein „combined artifact model of DT and RE“ zeigt, wie die verschiedenen DT und RE Artefakte zusammenwirken.

In vielen Seminarangeboten und Weiterbildungsveranstaltungen der letzten Jahre hat sich gezeigt, dass Softwareentwickler und Designer sich in „ihrer methodischen Welt“ sehr gut auskennen und dieses Wissen und die damit verbundene Erfahrung auch weiter nutzen wollen. Insofern sollte ein Integrationsvorschlag immer auf diesem bekannten Wissen aufbauen und den Wissenshorizont daran anknüpfend über die Disziplinengrenzen hinaus erweitern. Wenn sich dann ein neues Modell-Framework ergibt, ist der unmittelbare praktische Einsatz in aller Regel gewährleistet. Dieser Erkenntnis folgend soll im Folgenden eine „Modell-Integration“ vorgestellt werden, die sowohl Softwareingenieure als auch Designer abholt und sie in die Lage versetzt, danach in „synchronisierten“ mentalen Modellen koordiniert in einem Entwicklungsprojekt zusammenzuarbeiten.

## 2 VORGEHENSMODELLE

Es werden vier Vorgehensmodelle näher betrachtet: Zunächst wird die grundsätzliche Vorgehensweise im Requirements Engineering dargestellt, es folgt eine kurze Beschreibung der Menschenzentrierten Gestaltung (Human Centered Design). Kurze Ausführungen über Design Thinking schließen sich an. Den Abschluss bildet ein kurzer Abriss des sog. Usage Centered Design.

### 2.1 Requirements Engineering (grundsätzliche Vorgehensweise)

In der Produktentwicklung können die funktionale Ausgestaltung und die beide Qualitätsaspekte „Product Quality“ und „Quality in Use“ nicht getrennt werden. Die Qualität eines Produktes besteht darin, dass die richtigen Anforderungen ermittelt werden und dass alle Anforderungen, funktionale wie nicht funktionale Anforderungen im Produkt umgesetzt sind.

Ein typischer Anforderungsermittlungsprozess ist in Abbildung 1 dargestellt. Die Anforderungsermittlung beginnt mit dem Projektstart und gliedert sich in die folgenden grundsätzlichen Schritte (in Anlehnung an [4], [5]):

- (1) Ermittlung des Umfangs der insgesamt zu erbringenden Leistung (scope of the work)
- (2) Festlegung der Rolle eines Produktes, das bei der Leistungserbringung unterstützen kann (scope of the product)
- (3) Ableitung von Produkt-Anforderungen

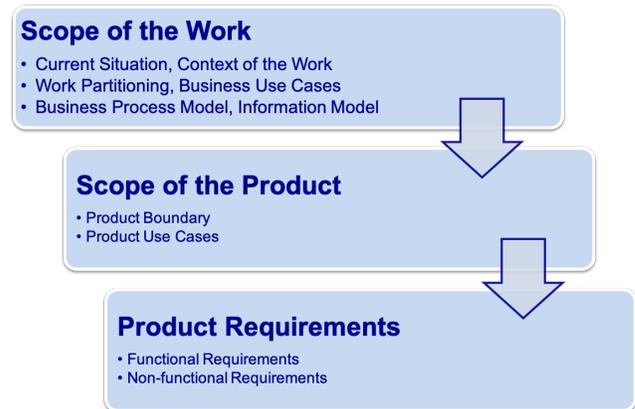


Abbildung 1: Requirements Engineering Prozess nach Roberstson (eigene Darstellung)

Die folgenden Abschnitte beschreiben diese Schritte im Detail.

- (1) Ermittlung des Umfangs der insgesamt zu erbringenden Leistung

Am Anfang steht die Ermittlung des Umfangs der zu erbringenden Leistung bzw. die Ermittlung der Aufgaben, die als Dienstleistung angeboten werden bzw. werden sollen. Man spricht von „scope of the work“, also vom Umfang, der insgesamt zu leistenden Arbeit. Es muss diese Arbeit verrichtet werden, um Leistungen (services) für den Kunden zu erbringen.

Üblicherweise beobachtet man dazu die Art, wie die angeforderte Leistung bisher erbracht wird und man beobachtet die Reaktion des bisherigen Systems. Die Beschreibung des Arbeitsbereichs umfasst die vorgesehenen Akteure (intended actors) und ggf. die Arbeitsleistung, die sie erbringen. Um die zu erbringenden Leistungen definieren zu können, muss klar sein, was an Unterstützung von externen Systemen (adjacent systems) zu erwarten ist, bzw. welche Rolle derartige Partnersysteme beim Erbringen der Dienstleistung spielen

können. Es muss dazu geklärt werden, welche technischen Fähigkeiten ein externes System hat, wie das externe System bei der Leistungserbringung bisher beteiligt ist bzw. welche Mitwirkung man sich für das externe System eventuell künftig vorstellen könnte.

Anschließend werden Geschäftsvorfälle und Geschäftsprozesse (Business Use Case), die in Zusammenhang mit dem Leistungsangebot bzw. der Aufgabenstellung stehen, identifiziert. Damit wird das Leistungsangebot strukturiert.

(2) Festlegung der Rolle des künftigen Produktes

Wie sollte ein Produkt aussehen, das möglichst optimal bei der Erbringung der beschriebenen Arbeitsleistung unterstützt? Bei diesen Überlegungen zu einem optimalen Produkt können unterschiedlichste Aspekte eine Rolle spielen, z.B. der Zweck, der mit dem Produkt verfolgt wird, die mögliche Mitwirkung externer Systeme, oder auch Auswirkungen der Produktdefinition auf Betroffene. Als Ergebnis dieser Überlegungen ergibt sich, welche Geschäftsprozesse ganz oder teilweise durch das Produkt unterstützt und im Produkt realisiert werden und welche Dienstleistungen weiterhin außerhalb des Produktes erbracht werden. Nun kann die so identifizierte Rolle des künftigen Produktes bei der Erbringung der Leistung durch Formulierung von Anwendungsfällen (Product Use Case) weiter strukturiert beschrieben werden.

(3) Ableitung von Anforderungen

Im letzten Schritt erfolgt die Ableitung von Anforderungen zu jedem Anwendungsfall. Nach dem beschriebenen Konzept entsteht eine Liste möglicher Anforderungen.

Es besteht die Gefahr, dass die nur in textueller Form vorhandenen Anforderungen von den verschiedenen Beteiligten ganz unterschiedlich interpretiert werden und ein auf dieser Basis entwickeltes Produkt doch nicht auf die volle Zustimmung des Kunden bzw. der Nutzer trifft. Aus diesem Grund sollte die Interpretation der Anforderungen in Bezug auf ein gemeinsames Verständnis überprüft werden. Am einfachsten gelingt dies durch eine prototypische Realisierung der potentiellen Anforderungen (requirements prototype). Durch die konkrete Umsetzung der Anforderungen in einem Prototypen wird den Stakeholdern wesentlich weniger Abstraktionsvermögen bei der Beurteilung der Anforderungen und ihrer Umsetzung abverlangt. Er kann seine Vorstellungen unmittelbar überprüfen. Mit diesem Prototypen konfrontiert, lassen sich schnell Missverständnisse ausräumen. Weiter bietet der Prototyp aber auch die Möglichkeit, von den Beteiligten fehlende Informationen zu erhalten und auf neue, andersartige und alternative Varianten in der Umsetzung aufmerksam zu werden. Eine durch einen Prototypen überprüfte Anforderungsliste führt zu einer konsolidierten Anforderungsspezifikation. Diese Anforderungsdefinition ist ein wichtiger Schritt auf dem Weg zu einem Qualitätsprodukt.

## 2.2 Menschzentrierte Gestaltung (Human Centered Design)

Der menschzentrierte Gestaltungsprozess (human centered design process) ist im Überblick in der Abbildung 2 dargestellt. Die einzelnen Phasen des menschzentrierten Gestaltungsprozesses werden im Folgenden vorgestellt.

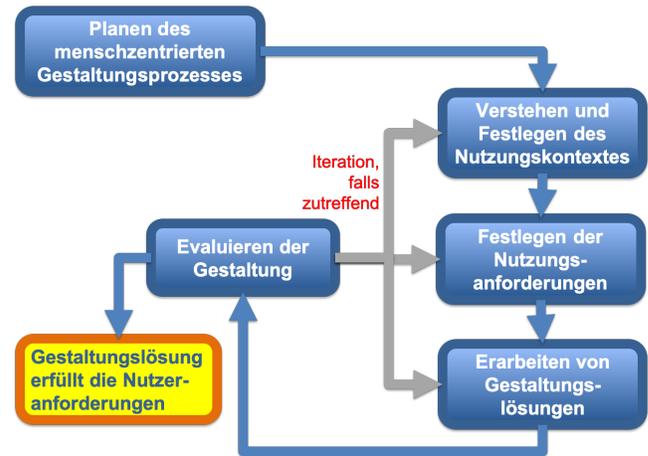


Abbildung 2: Human Centered Design Prozess gemäß ISO 9241 Teil 210 (2020) - eigene Darstellung

- (1) Planen des menschzentrierten Gestaltungsprozesses  
Die menschzentrierte Gestaltung muss für sämtliche Phasen des Produkt-Lebenszyklus geplant und in diese integriert werden.
- (2) Verstehen / Festlegen des Nutzungskontexts  
Es ist sinnvoll, Informationen zum aktuellen Nutzungskontext eines Produktes zu sammeln und zu analysieren, um das Umfeld der Nutzung (Ressourcen, Umgebungen) und die Nutzung selbst (Benutzer, Ziele, Aufgaben) besser zu verstehen.
- (3) Festlegen von Nutzungsanforderungen  
Bei den meisten Projekten stellt die Bestimmung der Erfordernisse der Benutzer als Basis für Anforderungen eine Hauptaktivität dar. Bei der menschzentrierten Gestaltung muss diese Aktivität erweitert werden, um eine klare Darstellung der Nutzungsanforderungen im Zusammenhang mit der Beschreibung des vorgesehenen Nutzungskontexts und der wirtschaftlichen Ziele des Systems. In Abhängigkeit vom Anwendungsbereich des Systems können die Nutzungsanforderungen Forderungen nach organisatorischen Änderungen und abgeänderten Arbeitsweisen umfassen. Nutzungsanforderungen könnten auch Vorschläge in Bezug auf Möglichkeiten zur Kombination von Produkten und Dienstleistungen unterbreiten.
- (4) Erarbeitung von Gestaltungslösungen  
Die nächste Phase im menschzentrierten Gestaltungsprozess beschäftigt sich mit dem Erarbeiten von Gestaltungslösungen zur Erfüllung der Nutzungsanforderungen. Dazu gehören das Gestalten der Benutzungsschnittstelle mit Benutzeroberfläche und der Benutzer-System-Interaktion. Die

menschzentrierte Gestaltung zielt in dieser Phase darauf ab, ein gutes Benutzererlebnis zu erreichen.

(5) Evaluieren von Gestaltungslösungen

Die nächste Phase im menschenzentrierten Gestaltungsprozess beschäftigt sich mit dem Evaluieren der Gestaltung anhand der Anforderungen aber auch in Bezug auf die grundsätzliche Konzeption des Produktes (sind die Annahmen zum Nutzungskontext angemessen, ...). Diese Überprüfung und Bewertung bietet die Möglichkeit ein besseres bzw. vertieftes Verständnis für die Erfordernisse der Benutzer zu gewinnen.

## 2.3 Design Thinking

Jochen Gürtler und Johannes Meyer (siehe [4]) beschreiben Design-Thinking so:

Design Thinking ist eine Arbeitsmethode, die verschiedene Werkzeuge verbindet, um Innovation und Ideenfindung zu unterstützen. ... Design Thinking kann helfen, Problemstellungen strukturiert und mit Spaß zu bearbeiten und zu Lösungsideen zu gelangen, die Sie und wir jetzt noch gar nicht kennen - echte Innovation eben.

und weiter:

Design Thinking versteht sich als Sammlung von Techniken verschiedener Disziplinen, die in Kombination die Erfolgswahrscheinlichkeit und Verlässlichkeit von nutzerzentrierten Ideen erhöhen können. Diese Definition macht deutlich, warum Design Thinking aktuell so große Bedeutung gewinnt: Man möchte innovative Entwicklungsergebnisse erzielen!

Man startet in dieser Arbeitsmethode mit einem „Innovationsauftrag“, also einer Beschreibung welches Problem der realen Welt neu bedacht werden soll. Man spricht von der „Design Challenge“. Die Problemfrage sollte möglichst allgemein formuliert sein, um viel Raum für neue Ideen freizugeben, sollte andererseits aber natürlich schon eine Richtung vorgeben, die zielorientiertes Arbeiten ermöglicht. Es gibt verschiedene Ausprägungen des Design-Thinking Prozesses. Diese verschiedenen Varianten unterscheiden sich nicht in der Zielsetzung jedoch in der Anzahl der Schritte, in denen man die innovative neue Lösung entwickelt. Hier soll eine Variante mit sechs Schritten vorgestellt werden. Die Abbildung 3 gibt einen Überblick über diesen sechsstufigen Design Thinking Prozess.

Die Schritte sind:

- (1) Initiale Phase des Verstehens
- (2) Beobachten / Erforschen / Empathie aufbauen
- (3) Standpunkt definieren
- (4) Ideen finden
- (5) Prototyp entwickeln
- (6) Testen

Die Schritte im Einzelnen:

- (1) Initiale Phase des Verstehens  
Ausgangspunkt ist die Problem-Frage (Design Challenge). Ziel der Phase ist es, das Problem an sich zu verstehen. Dazu sammelt man möglichst viele Informationen, die im Umfeld der Problem-Frage von Bedeutung sind: Welche Probleme gibt es, wie äußern sich die Probleme, wann treten sie auf,



Abbildung 3: Die sechs Schritte im Design Thinking Prozess nach Jochen Gürtler und Johannes Meyer - eigene Darstellung

wer ist davon betroffen? Ziel der Überlegungen in dieser Phase ist ein möglichst weit gefasstes Bild von Stakeholdern und ggf. möglichen Benutzern eines künftigen Produktes zu erhalten, das die aufgeworfenen Probleme löst.

(2) Beobachten / Erforschen / Empathie aufbauen

In dieser Phase steht im Vordergrund Verständnis für die Bedürfnisse und Probleme der tatsächlichen Nutzer zu entwickeln und den jeweiligen Nutzungskontext zu verstehen. Ziel der Beobachtungs-Phase ist es möglichst viel über das eigentliche Problem der Betroffenen von den Betroffenen zu lernen und diesmal im Unterschied zur ersten Phase die Informationen aus erster Hand zu bekommen. Dazu führt man Interviews und Beobachtungen von potentiellen Nutzern bzw. Stakeholdern durch.

(3) Standpunkt / Sichtweise definieren

In der Beobachtungsphase sind viele Ideen, wo Probleme bzw. Bedürfnisse liegen könnten, zusammengetragen worden. Nicht alle diese Ideen können weiterverfolgt werden. Aus der Vielzahl der gesammelten Informationen werden deshalb diejenigen extrahieren, auf die sich das Team in den folgenden Schritten konzentrieren will. Es wird jetzt festgelegt, welcher Aspekt des Ausgangsproblems, welche Problemsituation als erstes aufgegriffen und weiterbearbeitet werden soll. Methodisch kann das Entwickeln von Personas bei Verdichtung der eingesammelten Information sehr hilfreich sein.

(4) Ideen finden

In der Ideenfindungs-Phase geht es darum Ideen zum Lösen des ausgewählten Problems / der ausgewählten Probleme zusammen zu tragen. Das Ziel ist es möglichst viele unterschiedliche Lösungs-Konzepte zu entwickeln. In dieser Phase geht Quantität vor Qualität. Es kommen Kreativtechniken wie z.B. Brainstorming zum Einsatz.

(5) Prototyp entwickeln

Die groben Lösungsideen der letzten Phase müssen konkretisiert werden. Visualisierung der Lösungsidee in Form von Prototypen ist dazu ein taugliches Mittel. Prototypen machen die Lösungsidee „greifbar“, sie wird durch den Prototyp nachvollziehbar dargestellt und damit verständlich gemacht.

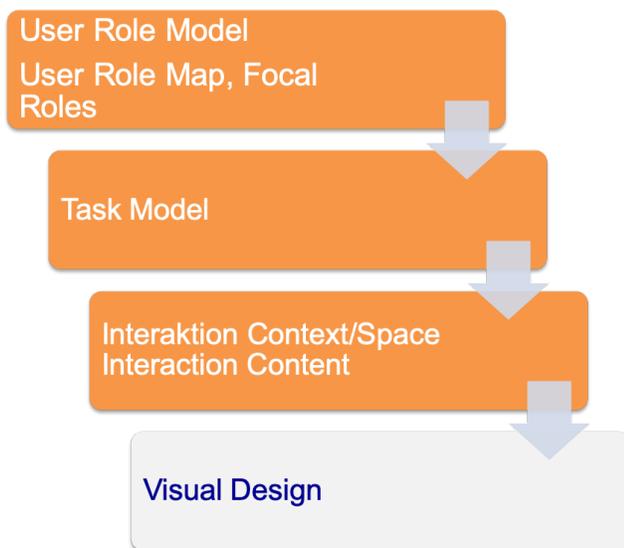
#### (6) Testen

Ziel in dieser Test-Phase ist es Erkenntnisse über die Benutzung, vor allem über Benutzungsprobleme zu erhalten und das Nutzer Feedback in der Weiterentwicklung der Lösung zu integrieren. Usability Evaluation, speziell Usability Tests kommen hier zum Einsatz.

Man kann das Wort „Design“ in Design-Thinking Prozess nicht nur auf die Fachdisziplin Gestaltung/Design beziehen, sondern durchaus sehr allgemein als Entwurfsprozess/Design einer System-/Produktentwicklung verstehen. Damit wird diese Methode auch für Entwickler anderer Fachdisziplinen außerhalb der Fachdisziplin Gestaltung/Design interessant.

## 2.4 Usage Centered Design

Wie entwickelt man systematisch die Benutzeroberfläche eines neuen interaktiven Produkts, einer neuen interaktiven Softwareanwendung? Einen systematischen Ansatz liefert eine Methode, die auf Constantine und Lockwood ([1]) zurückgeht, das sog. Usage Centered Design. Das Ziel dieser Methode ist, eine Oberflächenarchitektur für Anwendungen zu finden. Die Abbildung 4 gibt einen Überblick über die einzelnen Schritte.



**Abbildung 4: Usage Centered Design Prozess nach Constantine und Lockwood - eigene Darstellung**

Die vier Usage Centered Design -Schritte werden im Folgenden vorgestellt:

- (1) User und Rollen die User einnehmen (user and user roles): Natürlich werden wir uns jetzt speziell mit dem Benutzer beschäftigen, weil er der Ideengeber ist was die Benutzung

des Produktes betrifft. Der Benutzer wird dabei in seiner spezifischen Rolle wahrgenommen, in der er mit dem Produkt umgeht.

- (2) Arbeitsprozesse (working structures) und Aufgabenmodell (task model): Mit der Analyse von Arbeitsprozessen (working structures) identifiziert man welche Aufgaben Benutzer bearbeiten und wie sie von dem neuen Produkt dabei unterstützt werden können. Eine in sich abgeschlossene Aufgabe wird als Use Case (oft auch Task Case genannt) dargestellt. Ergebnis ist ein sogenanntes Task-Modell (task model). Um möglichst viele inhaltliche und strukturelle Ähnlichkeiten in verschiedenen Tasks zu erkennen, ist es hilfreich, Tasks (Task Cases) abstract als sog. „essential use cases“ zu formulieren.
- (3) Interfacearchitektur (interaction space, navigation map): Als ein Ergebnis des Usage Centered Design Prozesses wird die Benutzerschnittstelle als spezielle Komponente der Systemarchitektur abgeleitet. Die Interfacearchitektur beschreibt mit einer Definition des Interaktionsraums (interaction space), dem Festlegen der Inhalte der Interaktion (interaction content) und der Navigation zwischen Interaktionsräumen (navigation map) eine erste abstrakte Vorstellung über die Interaktion des Benutzers mit dem Produkt. Dieses Ergebnis wird auch als „abstract prototyp“ bezeichnet, weil die Oberflächenarchitektur ohne Konkretisierung einer technischen Umsetzung, auf abstrakte Weise beschrieben ist.
- (4) Visuelles Design (visual design): Im letzten Schritt wird dann das abstrakt abgeleitete Interaktionskonzept in eine konkret ausgestaltete und visualisierte Form gebracht.

## 3 DIE MODELL-INTEGRATION

Wie lassen sich die dargestellten disziplintypischen Vorgehensmodelle zu einem Modell für das „Human Centered Requirements Engineering“ integrieren?

Das **Usage Centered Design** Verfahren ist eine Konkretisierung des **Human Centered Design** Prozesses, wie er in EN ISO 9241 Teil 210 definiert ist. Der Nutzungskontext im Human Centered Design Prozess besteht aus den Komponenten Benutzer, Arbeitsaufgabe, Arbeitsmittel und Umgebung. Mit der ersten Usage Centered Design Phase wird der Benutzer und seine Rolle näher untersucht (User Role Model, User Role Map, Focal Roles). Damit trägt diese Phase zur Erschließung des Nutzungskontextes bei in dem Teilprozess „Verstehen und Festlegen des Nutzungskontextes“ im Human Centered Design Prozess. Die zweite Usage Centered Design Phase analysiert die zu lösenden Aufgaben (Task Model). Die entstehenden Aufgabenmodelle sind ebenfalls Elemente des Nutzungskontextes und tragen zum Teilprozess „Verstehen und Festlegen des Nutzungskontextes“ im Human Centered Design Prozess bei. Allerdings lassen Aufgabenmodelle (Task Model) des Usage Centered Design auch Rückschlüsse auf Anforderungen zu. Insofern sind diese Tätigkeiten im Usage Centered Design auch der Spezifikation von Benutzeranforderungen im Human Centered Design Prozess zuzuordnen (Teilprozess „Festlegen der Nutzungsanforderungen“). Die Festlegung von Interaktionsräumen (interaction context/space, interaction content) und das Konzipieren der Navigation (navigation map) liefern ebenfalls Benutzeranforderungen. Damit

ist auch diese Usage Centered Design Phase der Anforderungsspezifikation des Human Centered Design Prozesses im Teilprozess „Festlegen der Nutzungsanforderungen“ zuzuordnen. Die im abstract prototyp zusammengefasste Oberflächenarchitektur ist aber auch Basis für die Human Centered Design Phase „Erarbeiten von Gestaltungslösungen“.

Das **Usage Centered Design** Verfahren läßt sich auch sehr gut mit dem **Requirements Engineering** Prozess verknüpfen. Nachdem im Requirements Engineering Prozess das Produkt in seiner Leistungsausprägung (scope of the product) festgelegt wurde, wird die Produktleistung in Form von Product Use Cases für Akteure strukturiert erfasst. Akteure sind Benutzer in spezifischen Rollen, die spezielle Product Use Cases initiieren. Das Usage Centered Design kann durch sein User Role Model das Auffinden der „intended actors“ in der „scope of the work“- Analyse des Requirements Engineering unterstützen. Sowohl Requirements Engineering als auch Usage Centered Design leiten Use Cases bzw. Task Cases (tasks) ab. Hier ergänzen sich die beiden methodischen Ansätze. Diese konsolidierte Sicht auf Product Use Cases ist eine gute Basis für die im Usage Centered Design folgende Ableitung abstrahierter Use Cases (essential use case) als Grundlage für Überlegungen zu Interaktionsräumen (interaction space).

**Requirements Engineering** liefert auch wesentliche Informationen für den **Human Centered Design** Prozess. Mit den Überlegungen zu „scope of the work“ und „scope of the product“ wird das Festlegen des Nutzungskontextes im Human Centered Design Prozess wesentlich unterstützt. Die im Requirements Engineering abgeleiteten Anforderungen und die im Human Centered Design festgestellten Nutzungsanforderungen ergänzen sich.

Design Thinking bietet sich als methodische Klammer an. Das Requirements Engineering Vorgehen zeigt, dass es Aufgabe der Entwickler ist, nicht einfach ein bestehendes Produkt neu zu machen, sondern zu analysieren und Möglichkeiten zu erkennen, in welchem Arbeitsbereich ein neues Produkt eingesetzt werden soll. Die Analyse besteht nicht nur darin, die aktuell erbrachte Leistung möglichst gut zu verstehen, sondern auch darin, die eventuell beteiligten Partnersysteme hinsichtlich ihres Beitrags bei der Leistungserbringung wahrzunehmen und auch zu hinterfragen, inwieweit der bisherige Beitrag ggf. modifiziert, ersetzt oder weggelassen werden könnte. Es bedarf aber nicht nur der Analyse und des Verstehens der aktuellen Situation. Es ist auch im Blick der Requirements Engineers, wie sich die Arbeitssituation in der Zukunft gestaltet, und welche möglichen Leistungen dann berücksichtigt werden sollten. Hier ist sehr viel vorausschauende Kreativität nötig. Diese Aufgabe veranschaulicht das „Brown-Cow“-Modell (siehe [4]) in Abbildung 6. Ausgangspunkt ist die aktuelle Situation (Now), sowohl in der Analyse des gesamten Leistungsspektrums auf Basis von Geschäftsvorfällen / Business Use Cases (BUC) (Now-What) als auch in Bezug auf eine Produktversion in Anwendungsfällen / Product Use Cases (PUC) (Now-How). Die Verwendung der essentiell formulierten Use Cases erleichtert durch das Loslösen von konkreten Umsetzungsvorstellungen den Schritt zu innovativen neuen Use Cases. Die gegenwärtige Situation wird in die Zukunft extrapoliert: Künftige neue Leistungselemente (Future-What) werden auf künftige Produktumsetzungen (Future-How) projiziert.

Design Thinking bietet bei diesem „Blick in die Zukunft“ die methodische Unterstützung in den Phasen „verstehen“ und „beobachten“. Aus der Breite der Ideen muss aber wieder zu einer konkreten Vorstellung über die essentiellen Arbeitspakete gefunden werden, die für die weitere Produktentwicklung die Basis bilden. Die Design Thinking Phase „Standpunkt definieren“ unterstützt dieses Anliegen. Ergebnis ist die Definition der „essential work“.

Requirements Ingenieure entwickeln jetzt eine Vielzahl von Ideen, wie ein Produkt beim Erbringen der „essential work“ gestaltet werden könnte. Hier stellen sich z.B. Fragen nach Leistungsumfang und Automatisierungsgrad des künftigen Produkts. Aus der Palette der möglichen Produktrealisierungen (scope of the product) wird dann ein Favorit (selected product version) ausgewählt. Diese Aktivität ist der Design Thinking Phase „Ideen finden“ zugeordnet.

Diese ausgewählte Produktversion wird Basis für die weitere Entwicklungstätigkeit. Requirements Engineers entwickeln Product Use Cases aus denen sie funktionale Anforderungen ableiten. Wo sie sich unsicher sind, nutzen sie eine prototypische Implementierung um Klarheit über die Angemessenheit und Validität der Anforderung zu gewinnen. Dieser Prototyp kann jetzt auch dazu dienen, das Interaktionskonzept zu überprüfen. Dazu haben Usability Engineers den Nutzungskontext für die ausgewählte Produktversion ermittelt, Nutzungsanforderungen abgeleitet und auf dieser Basis den „Requirements Prototyp“ erweitert. Sinnvollerweise werden Interaktionsdesigner sich bei ihren Überlegungen auf die Product Use Cases beziehen. Damit ergibt sich ganz selbstverständlich, dass die Product Use Cases von Requirements Engineers und Interaktionsdesignern gemeinsam spezifiziert werden. Das Usage Centered Design kann jetzt als methodische Richtschnur Requirements Engineers und Interaktionsdesignern gleichermaßen dienen, um zu einem Produkt-Prototypen zu kommen. Das Usage Centered Design benötigt Input aus beiden Disziplinen, beide Disziplinen ergänzen sich beim Erstellen der Artefakte Interaktionsraum, Abstract Prototyp und Visual Design jeweils mit unterschiedlicher Schwerpunktsetzung. Hier werden die Design Thinking Phase „Prototyp entwickeln“ und die Human Centered Design Phase „Erarbeiten von Gestaltungslösungen“ durch das Usage Centered Design sehr gut methodisch erweitert.

Als letzter Schritt erfolgt in allen Vorgehensmodellen das Testen, das das Überprüfen getroffener Entwicklungsentscheidungen ermöglicht und durch das iterative Vorgehen des Human Centered Design Prozesses eine geeignete Überarbeitung des Prototypen bis hin zur Auswahl einer anderen Produktversion methodisch begleitet.

Das Zusammenwirken der verschiedenen Vorgehensmodell ist in der Abbildung 5 dargestellt.

## 4 ZUSAMMENFASSUNG

Design Thinking liefert einen methodischen Background und einen Vorgehensrahmen der das grundsätzliche Vorgehen im Requirements Engineering sehr gut unterstützt. Der menschenzentrierte Gestaltungsprozess bringt die Test-getriggerte inkrementelle Vorgehensweise in das Verfahren ein. Der Usage Centered Design Ansatz unterstützt die systematische Entwicklung der Oberflächenarchitektur ohne die kreativen Möglichkeiten des Interaktionsdesigns einzuschränken. Im vorgeschlagen integrierten Vorgehen findet

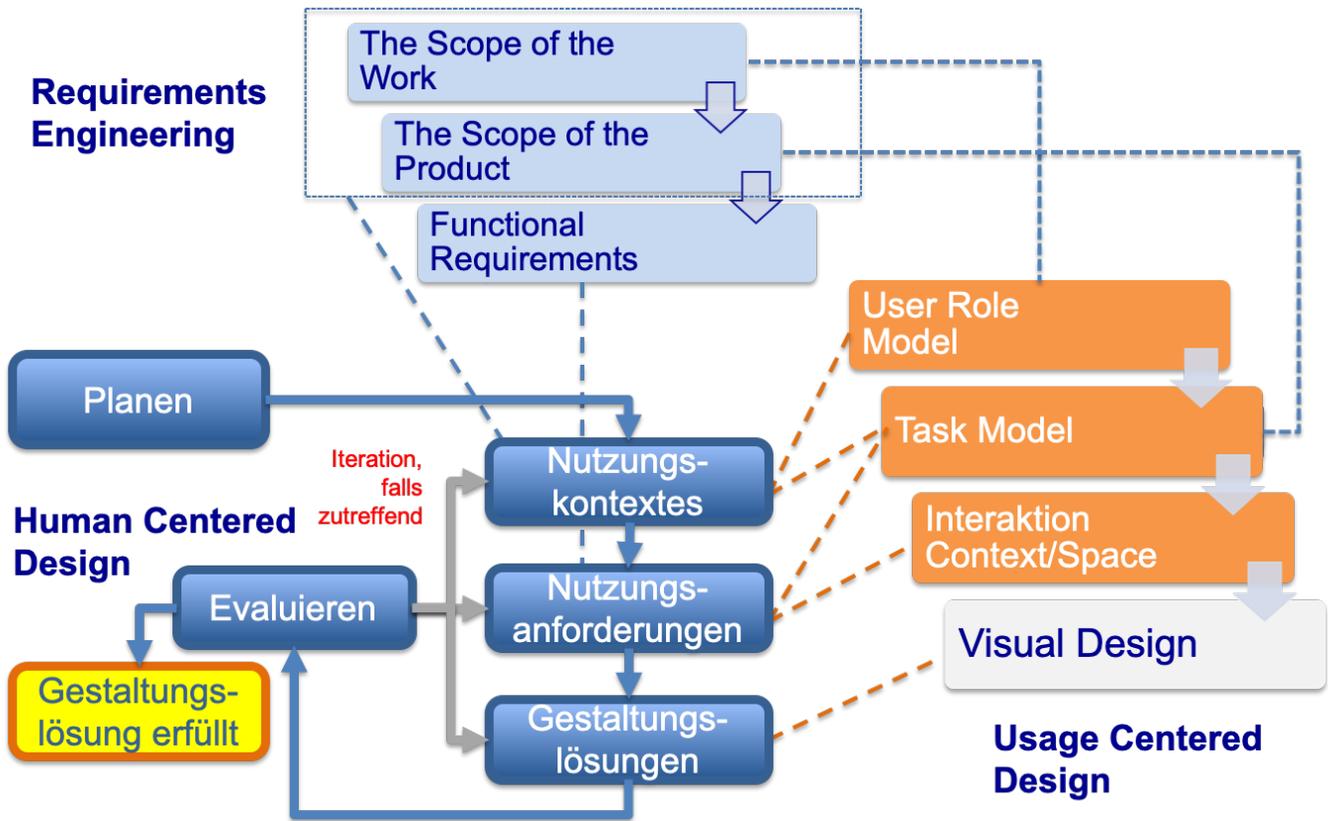


Abbildung 5: Das Zusammenspiel von Human Centered Design, Usage Centered Design und Requirements Engineering

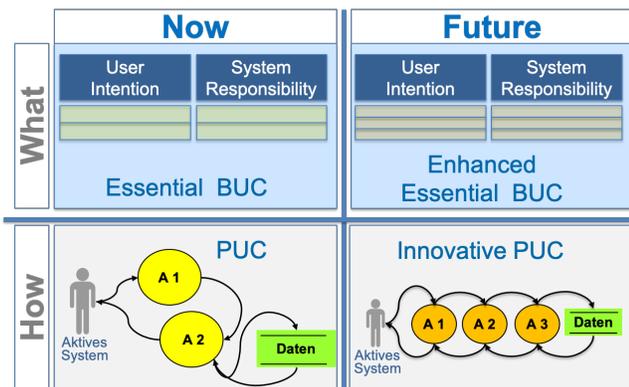


Abbildung 6: Künftige Anforderungen „erahnen“: Das „Brown-Cow“-Modell

jeder Produktentwickler, egal welcher fachlichen Disziplin er angehört, seine „Entwicklungsstrategie“ wieder und er lernt die methodische Expertise der anderen Fachlichkeiten als Bereicherung kennen. Nicht die Konkurrenz der Methoden, sondern die gegenseitige methodische Ergänzung und das gemeinsame Arbeiten am Produkt stehen im Vordergrund.

### LITERATUR

- [1] Larry L. Constantine and Lucy A. D. Lockwood. 1999. *Software for use: A practical guide to the models and methods of usage-centered design*. Addison Wesley, Reading, Mass. <http://proquest.tech.safaribooksonline.de/9780768685305>
- [2] Jennifer Hehn, Daniel Mendez, Falk Uebernickel, Walter Brenner, and Manfred Broy. 2020. On Integrating Design Thinking for Human-Centered Requirements Engineering. *IEEE Software* 37, 2 (2020), 25–31. <https://doi.org/10.1109/MS.2019.2957715>
- [3] Pardha S. Pyla, Manuel A. Pérez-Quinones, James D. Arthur, and H. Rex Hartson. 2005. Ripple: An Event Driven Design Representation Framework for Integrating Usability and Software Engineering Life Cycles. In *Human-Centered Software Engineering – Integrating Usability in the Software Development Lifecycle*, John Karat, Jean Vanderdonck, Gregory Abowd, Gaëlle Calvary, John Carroll, Gilbert Cockton, Mary Czerwinski, Steve Feiner, Elizabeth Furtado, Kristiana Höök, Robert Jacob, Robin Jeffries, Peter Johnson, Kumiyo Nakakoji, Philippe Palanque, Oscar Pastor, Fabio Paternò, Costin Pribeanu, Marilyn Salzman, Chris Schmandt, Markus Stolze, Gerd Szwillus, Manfred Tscheligi, Gerrit van der Veer, Shumin Zhai, Ahmed Seffah, Jan Gulliksen, and Michel C. Desmarais (Eds.). Human-Computer Interaction Series, Vol. 8. Springer Netherlands, Dordrecht, 245–265. [https://doi.org/10.1007/1-4020-4113-6\\_13](https://doi.org/10.1007/1-4020-4113-6_13)
- [4] Suzanne Robertson and James Robertson. 2013. *Mastering the requirements process: Getting requirements right* (3rd ed. ed.). Addison-Wesley, Upper Saddle River, N.J. <http://proquest.tech.safaribooksonline.de/9780132942850>
- [5] Suzanne Robertson and James Robertson. 2018. *Volere Requirements Specification Template*. Technical Report. Atlantic Systems Guild. Edition 18, 2016.
- [6] Ahmed Seffah, Michel C. Desmarais, and Jan Gulliksen. 2005. *Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle*. Human-Computer Interaction Series, Vol. 8. Springer, Dordrecht. <https://doi.org/10.1007/1-4020-4113-6>