

Universität Konstanz

FB Informatik und Informationswissenschaft

Master-Studiengang Information Engineering

Masterarbeit

Förderung eines benutzerzentrierten Software-Entwicklungsprozesses in Unternehmen am Beispiel von UBS Schweiz

**zur Erlangung des akademischen Grades eines
Master of Science (M.Sc.)**

Vorgelegt von

Siegrun Annette Heberle, Matr.Nr. 01 / 464668

Studienfach:
Schwerpunkt:
Themengebiet:

Information Engineering
Information Science
Angewandte Informatik

Erstgutachter und Betreuer:
Zweitgutachter:
Eingereicht am:

Prof. Dr. Harald Reiterer
Prof. Dr. Rainer Kuhlen
23. Mai 2002

Förderung eines benutzerzentrierten Software-Entwicklungsprozesses in Unternehmen am Beispiel von UBS Schweiz

„It is difficult for a company to make the transition to a consumer-driven marketplace. It requires an entirely new approach to products, starting with an understanding of customers and their needs, their real needs, not the feature sets so loved by marketing. It means using social scientists to collect, analyze, and work with the data – moreover, social scientists on an even footing with engineers and technologists. It means structuring the whole product process differently than before. It may well mean reorganizing the company.“

(Norman 1998, S. 206)

Dank

Dass es zu dieser Masterarbeit überhaupt gekommen ist, verdanke ich in erster Linie meinem Mann, Stefano Simoni, dessen Ungeduld über meine berufliche Unzufriedenheit bei mir Berge versetzt hat. Ihm gehört nicht nur deswegen mein grösster Dank.

Ausserdem bedanken möchte ich mich bei meinem Professor, Harald Reiterer, der mir einerseits alle Freiheit zukommen liess und mir andererseits mit vielen Tipps, Hinweisen und konstruktiven Diskussionen zur Seite stand. Herzlichen Dank.

Meine Kollegen aus dem Usability Engineering bei der UBS haben alleine durch ihre Tätigkeiten, ihre Erfahrungen, die sie mit mir geteilt haben und ihre Bereitschaft, mit mir „meine Themen“ zu diskutieren, vermutlich mehr zu dieser Arbeit beigetragen, als ihnen bewusst ist. Auch ihnen herzlichen Dank. Besonderer Dank gilt hierbei Christian Hübscher, der mich mit seinen zahlreichen „Ich dachte, das könnte noch was für dich sein...“ E-mails mit vielen wertvollen Literatur- und sonstigen Hinweisen eingedeckt hat und immer spannend fand, was ich ihm zu erzählen hatte. Ich hoffe, dass dies noch lange so bleiben wird. Weiterhin möchte ich auch allen anderen Kolleginnen und Kollegen bei der UBS danken, die mir unbürokratisch alle gewünschten Informationen zukommen liessen.

Bedanken möchte ich mich auch bei all meinen Freunden, die in den letzten drei Jahren akzeptiert haben, dass ich so wenig Zeit hatte, und die trotzdem noch „mit mir spielen“ wollen.

Vorausgehende Bemerkungen

Alle Einschätzungen und Wertungen bezüglich UBS Schweiz entsprechen meiner persönlichen Meinung und sind in gewissem Masse subjektiv. Sie beruhen auf eigenen Erfahrungen im Zusammenhang mit der Beratung von Entwicklungsprojekten, zahlreichen Gesprächen mit Kollegen und Projektpartnerinnen sowie Beobachtungen während der letzten anderthalb Jahre.

Siegrun Heberle Simoni
Hardturmstrasse 130
CH-8005 Zürich

Telefon: +41-1-271 87 47
Email: sheberle@bluewin.ch

Verwendete Terminologie

Geschlechtsspezifische Begriffe: Um dem Denken in männlichen Prototypen von Entwicklern, Computerbenutzern, Designern usw. einen kleinen Riegel vorzuschieben und dennoch die Leserlichkeit des Textes nicht unnötig zu erschweren, wird in dieser Arbeit das Geschlecht abgewechselt. So wird, sofern keine konkreten männlichen oder weiblichen Personen angesprochen sind, z.B. einmal vom «Benutzer» ein anderes Mal von der «Benutzerin» die Rede sein. Eine Ausnahme von dieser Regelung stellt – wiederum aus Gründen der Leserlichkeit, aber auch zur Unterstützung einer Begriffsbildung – das Wort «Benutzerzentrierung» dar, das immer in dieser Form verwendet wird.

Benutzerzentriertheit: Der Begriff «Benutzerzentriertheit» wird in dieser Arbeit als Entwicklungsparadigma verwendet, das die konsequente Einbeziehung von Benutzerinnen in den Entwicklungsprozess bezeichnet. Dies ist jedoch keinesfalls damit zu verwechseln, die Benutzer zu Designern zu machen oder Benutzerinnen die Entscheidung über den Funktionalitätsumfang eines Softwareproduktes zu überlassen. Mit Benutzerzentriertheit ist gemeint, dass durch *systematische* Berücksichtigung der Benutzercharakteristika, der Benutzeraufgaben und –ziele sowie des Benutzungskontextes ein Produkt entwickelt wird, das maximale Usability aufweist.

Usability: Der Begriff «Usability» beschreibt die Qualität einer interaktiven Software oder eines Computersystems im Hinblick auf einfache Erlernbarkeit, einfache Bedienbarkeit und Benutzerzufriedenheit. In dieser Arbeit wird «Usability» in der umfassenden Bedeutung der Nutzungsqualität verwendet und ist damit abgegrenzt zum Begriff «Softwareergonomie» der hier in der begrenzteren Bedeutung bezogen auf die Ausgestaltung der Benutzerschnittstelle gebraucht wird (eigentliche Produktqualität). Demnach kann also eine Benutzerschnittstelle durchaus ergonomisch sein, ohne deswegen zwangsläufig über eine hohe Usability zu verfügen.

Im Deutschen existieren u.a. die Varianten «Benutzerfreundlichkeit», «Benutzungsfreundlichkeit», «Gebrauchstauglichkeit» und «Anwenderfreundlichkeit». Da in der deutschsprachigen Fachwelt noch weitgehend Uneinigkeit herrscht, wie der Begriff nun korrekt zu übersetzen ist, wird in dieser Arbeit der englische Begriff «Usability» verwendet. In der Verwendung als Adjektiv wird das deutsche Wort gebrauchstauglich beibehalten.

Usability Engineering (UE): Die systematische Anwendung von Methoden zur Erreichung einer hohen Nutzungsqualität eines Produktes wird als Usability Engineering bezeichnet. Usability Engineering verfolgt die Prinzipien Benutzereinbeziehung, Festlegung von überprüfbaren Usability Zielen, iteratives Design und wiederholte Evaluation zusammen mit Benutzerinnen. Da Benutzerzentriertheit in der Regel durch die Anwendung von Usability Engineering Methoden realisiert wird, werden die Begriffe «Benutzerzentriertheit» und «Usability Engineering» z.T. synonym verwendet, auch wenn dies formal nicht ganz korrekt ist. Vor allem im Zusammenhang mit Expertise und Organisationseinheiten ist der Begriff «Benutzerzentrierung» nur schlecht oder gar nicht brauchbar, weshalb hier z.B. mit dem Begriff «UE-Experten» all jene bezeichnet werden, die in irgendeiner benutzerzentrierten Aktivität die benötigte Expertise nachweisen können, ohne konkret zu differenzieren, ob es sich z.B. um Expertinnen im Sammeln von Benutzeranforderungen, im Erstellen von konzeptuellen Modellen oder im Ausgestalten von Benutzeroberflächen nach ergonomischen Gesichtspunkten handelt.

Prozess und Prozessmodell vs. Methode: Der Begriff «Prozess», wie er hier verwendet wird, beschreibt einen konkreten Ablauf zur Erreichung eines bestimmten Zieles (z.B. zur Entwicklung eines Software Produktes → Entwicklungsprozess). Der in dieser Arbeit verwendete Begriff «Prozessmodell» beschreibt einen theoretischen, idealtypischen Prozess, der in Form eines realen Prozesses instanziiert werden muss. «Methode» beschreibt in dieser Arbeit die Anleitung zur Durchführung einer bestimmten Aktivität im Rahmen eines Prozesses (z.B. Aufgabenanalyse). Die in anderen Zusammenhängen anzutreffenden Begriffe «Methode», «Methodologie» oder «Vorgehensmodell», die z.T. ebenfalls den kompletten Ablauf zur Erreichung eines bestimmten Zieles (z.B. der Usability des Endproduktes) bezeichnen, wurden in der vorliegenden Arbeit nur im Zusammenhang mit Zitaten und stehenden Bezeichnungen (z.B. Delta Methode) beibehalten. Im übrigen wurden sie durch die Begriffe «Prozess» bzw. «Prozessmodell» ersetzt.

Glossar: Alle in dieser Arbeit verwendeten Begriffe, die vermutlich nicht selbsterklärend sind, werden im Glossar definiert. Im Text sind diese Begriffe kursiv dargestellt und bei der ersten Verwendung mit einem Hyperlink auf den Glossareintrag versehen.

Kurzzusammenfassung

Warum gibt es bis heute so viele Softwareprodukte, die eine so schlechte Usability aufweisen? Warum lassen gerade grosse Firmen, die intern Software entwickeln, dieses für uns so offensichtliche Potential zur Steigerung der Effizienz, Effektivität und Zufriedenheit ihrer Mitarbeiterinnen brachliegen? Diese Arbeit zeigt auf, dass alle notwendigen Mittel zur Entwicklung eines gebrauchstauglichen Software-Produktes in der Theorie und teilweise sogar in der Praxis vorhanden sind. Das Problem liegt also nicht in der fehlenden Lösung, sondern vielmehr in einem fehlenden Bewusstsein von der Lösung, und – als noch viel grösseres Hindernis – in einem fehlenden Bewusstsein vom Problem selbst.

Betrachtet man Benutzerzentrierung als Innovation und den Prozess dorthin als organisationelles Lernen, ergeben sich interessante Aspekte, die nützlich sein können, um zukünftig benutzerzentrierte Prozesse zu fördern.

Anhand eines Reifegradmodells, das sich an vorhandenen Regelwerken aus dem Software- und Usability Engineering orientiert, wird in dieser Arbeit die optimal benutzerzentrierte Entwicklungsorganisation entworfen. Das Modell unterscheidet sieben Dimensionen: Neben (1) dem eigentlichen Entwicklungsprozess sowie (2) den Methoden und Tools werden (3) die Organisationsstruktur, (4) die Qualifikation der Mitarbeiterinnen, (5) das Schulungswesen und Wissensmanagement, (6) Managementaktivitäten und Kultur und nicht zuletzt (7) die verfügbaren Ressourcen betrachtet. Die Reifegrade deuten an, wie der Weg dorthin aussehen könnte. Am Beispiel von UBS Schweiz wird die Anwendung des Reifegradmodells demonstriert und gezeigt, dass das Modell dazu dienlich ist, neue Bereiche aufzuzeigen, in denen strategische Aktivitäten zur Förderung der Benutzerzentrierung notwendig sind. In diesem Zusammenhang werden auch die neuesten Ansätze zur technologischen Unterstützung eines organisationellen Lernprozesses im Hinblick auf Usability Engineering Aktivitäten vorgestellt.

Abstract

Why are we still confronted with such a poor usability in many software products? What lets all those big companies ignore the obvious opportunity to increase their efficiency, effectiveness, and employee satisfaction? This master thesis demonstrates that we have all required means in both, theory and practice, to develop usable software. Hence, the problem is not the missing solution but the missing awareness of the solution, and, even worse, the missing recognition of the problem itself.

Considering user-centredness as innovation and the process to its implementation as organisational learning delivers useful indications to further strategic efforts on the way to user-centred development processes.

Following the ideas of several widely distributed software engineering and usability engineering standards this thesis presents a maturity model of an ideal user-centred organisation. The model distinguishes seven dimensions: (1) the development process, (2) methods and tools, (3) the organisational structure, (4) employees' qualification, (5) the education system and knowledge management, (6) management activities and culture, and (7) resources; three maturity levels lead the way to real user-centredness.

The applicability of the maturity model is demonstrated in a case study about UBS Switzerland. The case study shows the benefit of using a model to support analysis and to direct strategic considerations with regard to user-centredness. Finally, the latest technological approaches to usability engineering and organisational learning are discussed.

Inhalt

1	Einleitung	1
1.1	Ziele und Aufbau der Arbeit	2
2	Software-Entwicklungsprozesse unter der Lupe	4
2.1	Qualitätsmanagement nach ISO 9001 und ISO 9000-3	4
2.2	Die Prozessqualität	4
2.2.1	Das Capability Maturity Model	4
2.2.2	ISO 15504 (SPICE)	5
2.2.3	Benutzerzentrierung: ISO 13407, DATech Prüfbaustein und Usability Maturity Model	7
2.2.4	Das Organisationelle Umfeld: Usability Maturity Model: Human Centredness Scale	9
2.3	Die Produktqualität	9
2.3.1	Allgemeine Produktqualität: ISO 9126-1	10
2.3.2	Die Usability des Produktes: ISO 9241 (Teil 10 – 17) und ISO 12119	10
2.3.3	Die Produktprüfung: DATech-Prüfhandbuch Gebrauchstauglichkeit	10
2.4	Zusammenfassung	11
3	Usability Engineering Prozesse in Theorie und Praxis	13
3.1	Charakterisierung der bekanntesten Usability Engineering Vorgehensweisen	13
3.1.1	Der Usability Engineering Lifecycle (Mayhew 1999)	13
3.1.2	Delta Method (WM-data AB & Ericsson Radio Systems AB 2000)	16
3.1.3	Contextual Design (Beyer und Holtzblatt 1998)	18
3.1.4	Scenario-based Development (Rosson und Carroll 2002)	21
3.1.5	Usage-Centered Design (Constantine und Lockwood 1999)	24
3.2	Probleme bei der praktischen Umsetzung von Usability Engineering Prozessen	27
3.3	Zusammenfassung	29
4	Benutzerzentrierung als Innovation	30
4.1	Der Innovationsprozess in Organisationen	31
4.2	Attribute von Innovationen	32
4.3	Innovationsfreudigkeit einer Organisation	33
4.4	Innovationsfreudigkeit eines Individuums	34
4.4.1	Motivation	34
4.4.2	Wahrnehmung von Rollen	34
4.5	Zusammenfassung	35
5	Die benutzerzentrierte Entwicklungsorganisation – Ein Reifegradmodell	36
5.1	Dimensionen einer benutzerzentrierten Entwicklungsorganisation	37
5.1.1	Ein Prozessmodell für benutzerzentrierte Software-Entwicklung	38
5.1.2	Das Angebot geeigneter Usability Engineering Methoden und Werkzeuge	43
5.1.3	Die passende Organisationsstruktur	44
5.1.4	Managementaktivitäten und Organisationskultur als kritische Erfolgsfaktoren	45
5.1.5	Qualifizierte Usability Engineering Mitarbeiter und Mitarbeiterinnen	49

5.1.6	Ein leistungsfähiges Schulungswesen und Wissensmanagement	50
5.1.7	Ausreichende Ressourcen für die übergeordneten Aufgaben	51
5.2	Das Ratingverfahren	53
5.2.1	3 Reifestufen für jede Dimension	53
5.3	Zusammenfassung	57
6	Das Fallbeispiel UBS Schweiz.....	59
6.1	Software-Entwicklung bei UBS Schweiz	60
6.1.1	Die GB IT Value Chain	60
6.1.2	Usability Engineering bei UBS Schweiz	61
6.1.3	Relevante Einflussgrößen: Corporate Center und Marketing	62
6.2	Reifegrad der Benutzerzentrierung bei UBS Schweiz	63
6.2.1	Die Entwicklungsrichtlinien (ERL) von UBS Schweiz als Standardprozessmodell	63
6.2.2	Das Angebot an Methoden und Tools zur benutzerzentrierten Software-Entwicklung	65
6.2.3	Benutzerzentrierung und Organisationsstruktur	66
6.2.4	Managementaktivitäten und Kultur zur Förderung von Benutzerzentrierung	67
6.2.5	Die Qualifikation der Mitarbeiterinnen im Hinblick auf benutzerzentrierte Aktivitäten	69
6.2.6	Unterstützung von Benutzerzentrierung durch Schulungswesen und Wissensmanagement.....	70
6.2.7	Ressourcen zur Förderung von benutzerzentrierter Software-Entwicklung	70
6.2.8	Zusammenfassende Bewertung der Benutzerzentriertheit von UBS Schweiz	72
6.3	Zusammenfassung	74
7	Massnahmen zur Erhöhung der Benutzerzentriertheit von UBS Schweiz	75
7.1	Strategische Massnahmen	75
7.2	Infrastrukturmassnahmen	80
7.2.1	Tools zur Unterstützung des Designprozesses	81
7.2.2	Tools zur Optimierung des Entwicklungsprozesses	82
7.2.3	Bewertung und Umsetzbarkeit der toolbasierten Ansätze bei UBS Schweiz	84
7.3	Zusammenfassung	85
8	Erfahrungen und Ausblick	86
9	Glossar	88
10	Literaturverzeichnis	90
11	Anhang	I
11.1	Vergleich der verschiedenen Usability Engineering Prozessmodelle	I

Abbildungsverzeichnis

Abbildung 1: Die 5 Reifegradstufen des CMM	5
Abbildung 2: Bestandteile von SPICE	6
Abbildung 3: Prozesskategorien in SPICE / ISO 15504	6
Abbildung 4: Normen und Prozessmodelle in der Software-Entwicklung	11
Abbildung 5: Der Usability Engineering Lifecycle	15
Abbildung 6: Prozess der Delta Method	17
Abbildung 7: Prozess des Contextual Designs.....	20
Abbildung 8: Überblick über ein Prozessmodell auf der Basis von Scenarios.....	23
Abbildung 9: Zusammenhang zwischen Modellen des Usage-Centered Designs.....	24
Abbildung 10: Modell eines Innovationsprozesses in Organisationen	31
Abbildung 11: Faktoren für die Innovationsfreudigkeit von Unternehmen	34
Abbildung 12: Dimensionen von Benutzerzentrierung	38
Abbildung 13: Zusammenhang zwischen Prozesskategorien, Zielen und Aktivitäten	40
Abbildung 14: Modell zur Beurteilung von Benutzerzentrierung	58
Abbildung 15: Organisationsstruktur des UBS Konzerns	59
Abbildung 16: Die GB IT Value Chain	61
Abbildung 17: Die Verwendung und Modifikation von Usability Quellen.....	82
Abbildung 18: Die Komponenten von ProUSE	83

Tabellenverzeichnis

Tabelle 1: Anforderungen der 7 Dimensionen an jede Reifegradstufe	54
Tabelle 2: Reifegrad des Prozessmodells von UBS Schweiz hinsichtlich Benutzerzentrierung.....	63
Tabelle 3: Qualität des Methoden- und Toolangebots.....	65
Tabelle 4: Unterstützung der Benutzerzentrierung durch eine geeignete Organisationsstruktur	66
Tabelle 5: Eignung von Managementaktivitäten und Kultur zur Förderung der Benutzerzentrierung	67
Tabelle 6: Stand der Qualifikation der Mitarbeiter	69
Tabelle 7: Reifegrad des Schulungswesens und Wissensmanagements bezüglich Benutzerzentrierung.....	70
Tabelle 8: Ressourcenverfügbarkeit zur Förderung von Benutzerzentrierung.....	70
Tabelle 9: Zusammenfassende Bewertung der Benutzerzentriertheit von UBS Schweiz.....	72
Tabelle 10: Strategische Massnahmen zur Erhöhung der Benutzerzentriertheit bei UBS Schweiz	76
Tabelle 11: Vergleich von 5 bekannten Usability Engineering Prozessmodellen	I

1 Einleitung

Nach über 30 Jahren softwareergonomischer Forschung und Praxis werden wir nach wie vor mit sehr schlechten Produkten im Sinne fehlender Usability konfrontiert. Dies muss nicht zuletzt als eine Folge ungeeigneter Entwicklungsprozesse angesehen werden. Die Kluft zwischen Usability Engineering (UE) und klassischen Software-Engineering (SE) Prozessen besteht nach wie vor, doch während Software-Engineering Aktivitäten in vielen Bereichen zur Pflicht geworden sind (z.B. benötigte Zertifizierung von Entwicklungsprozessen für Aufträge der Öffentlichen Hand in Deutschland), ist eine derartige Regelung für Usability-Belange trotz ISO 13407 und klaren Anforderungen an die Usability von Produkten, z.B. ISO 9241, Teile 10 und 11, noch nicht in Sicht.

Sehr viele Software-Entwicklungsprojekte scheitern. Man geht davon aus, dass nur ca. 1/3 aller Entwicklungsprojekte erfolgreich im geplanten Zeit- und Kostenrahmen abgeschlossen werden. Der Grund, dass so häufig der geplante Kostenrahmen überschritten wird, lässt sich in der überwiegenden Anzahl der Fälle auf eine unsaubere Anforderungsspezifikation zurückführen. Dies führt dann dazu, dass ganze Aufgabenbereiche übersehen werden, dass ständig neue Anforderungen nachgeschoben werden und dass aufgrund der mangelhaften Problemdefinition auch nur ein mangelhaftes Produkt entsteht, das zudem teurer wird als ursprünglich veranschlagt. (Lederer und Prasad 1992)

Grundsätzlich kann man davon ausgehen, dass sowohl die notwendigen Aktivitäten als auch die entsprechenden Methoden und Vorgehensweisen zu deren Durchführung vorhanden sind, um die genannten Usabilityprobleme in den Griff zu bekommen. In der umfangreichen Usability Engineering Literatur werden Prozessmodelle unterschiedlichster Ausprägung vorgestellt, die allesamt Antworten geben auf die Frage, wie es gelingen kann, die Bedürfnisse der zukünftigen Benutzer eines Software-Produktes besser zu verstehen und in Form einer gebrauchstauglichen Software zu bedienen. Auch wenn der konkrete, quantifizierbare Nachweis der Effizienz und Effektivität solcher Methoden noch aussteht, sprechen die zahlreichen Erfahrungsberichte von Usability Expertinnen dafür, dass Usability Engineering Vorgehen tatsächlich geeignet sind, die Nutzungsqualität von Software-Produkten zu erhöhen.

Nur unbefriedigend beantwortet ist bis anhin jedoch die Frage, warum die vorhandenen Vorgehensweisen des Usability Engineerings nur so spärlich zur Anwendung kommen. Aus Sicht des Projektmanagements kann man bemerken, dass das Führen eines UE-Prozesses **parallel** zu einem bereits eingeführten Software-Entwicklungsprozess nicht praktikabel ist, da dies doppelte Dokumentenführung bedingt und die Projektzuständigkeiten unnötig aufsplittet – mit allen negativen Konsequenzen, wie z.B. dem Kampf um Ressourcen. Da zudem oftmals ein Interessenkonflikt zwischen Usability Engineering und Software Engineering wahrgenommen wird, ist dieser Ansatz eher kontraproduktiv! Doch auch die Integration beider Prozesse ist oft schwierig, da:

1. Die Prozesse z.T. schwer vereinbar sind (z.B. Wasserfall vs. Iteration);
2. Klassische Methodenorientierung der UE-Prozesse entweder nicht akzeptiert wird oder das nötige Know-how zur erfolgreichen Durchführung fehlt.
3. Das Unternehmen die Bedeutung von Usability nicht erkennt oder das Thema nicht ernst genug nimmt (z.B. wenn das bloße Vorhandensein eines Usability Labors als ausreichend betrachtet wird).
4. Die Integration von UE in den Standard-Entwicklungsprozess eines Unternehmens einen längeren Lernprozess im Unternehmen bedeutet.

Bei diesen Überlegungen wird deutlich, dass das Problem weit über den eigentlichen Entwicklungsprozess hinausgeht. Ob eine Umorientierung in Richtung Benutzerinnen und Usability möglich wird, hängt nicht zuletzt auch von den organisatorischen Rahmenbedingungen ab, unter denen Software entwickelt wird. Hierzu gehören neben dem Aufbau eines entsprechenden Bewusstseins und Know-hows sowie der Errichtung einer geeigneten Organisationsstruktur auch die entsprechende Kultur und Werthaltung innerhalb einer Entwicklungsorganisation, ohne die das Bemühen um gebrauchstaugliche Produkte nicht gefördert werden kann.

Dass dieses Thema häufig von Usability Fachleuten aufgegriffen und verfolgt wird, ist kein Zufall. Schliesslich stellt die Abhilfe schlechter Usability von Software-Produkte deren Existenzberechtigung dar. Dennoch darf man nicht vergessen, dass Usabilityexpertinnen keinen Exklusivanspruch auf das Thema Benutzerzentrierung haben. Angesichts des Problemumfangs sollte daher ein Hauptaugenmerk der Bemühungen darauf liegen, im Kampf gegen schlechte Benutzerschnittstellen Verbündete zu suchen. Diese finden sich in

Marketingabteilungen, unter Designerinnen und – unter den Entwicklern selbst! Ein wichtiger Erfolgsfaktor auf dem Weg zur benutzerzentrierten Organisation ist es, gemeinsam mit allen Stakeholdern der Software-Entwicklung an einem Strick zu ziehen und dabei nicht aus den Augen zu verlieren, dass es aus allen Funktionsbereichen berechnete (Geschäfts-)Ziele und Interessen gibt, auch wenn diese nicht immer mit den Vorstellungen der Usabilityriege in Einklang stehen. (Anderson 2000)

1.1 Ziele und Aufbau der Arbeit

Im Rahmen dieser Arbeit wird ein Konzept zur Förderung der Benutzerzentrierung von Software-Entwicklungs-Organisationen vorgestellt. Das Konzept basiert auf Standardprozessen des Software- und Usability Engineerings, die im Überblick dargestellt und untersucht werden.

Unzulänglichkeiten der vorhandenen Prozesse bezüglich ihrer erfolgreichen Umsetzung werden aufgezeigt und analysiert. Anschliessend wird versucht, unter Ausschöpfung grundlegender Ideen der Innovationstheorie, die strategischen Schwächen der vorhandenen Software- und Usability Engineering Modelle zu überwinden. Hierzu werden beispielsweise die Dimensionen Organisationsstruktur sowie Managementaktivitäten und Kultur eingeführt, die gleichberechtigt neben der Dimension des eigentlichen Prozessmodells betrachtet werden. Das Modell soll sowohl die Analyse der Ist-Situation innerhalb einer Software-Entwicklungsorganisation erlauben als auch konstruktive Hinweise zu deren Verbesserung liefern.

Anhand eines Fallbeispiels soll die praktische Brauchbarkeit des Modells gezeigt werden, indem

- die Organisationsstruktur und der Stand der Benutzerzentrierung bei UBS Schweiz untersucht und bewertet wird;
- auf der Basis der Evaluation Massnahmen zur Verbesserung der Benutzerzentrierung bei UBS Schweiz formuliert werden;
- diese Massnahmen auf ihre Umsetzbarkeit hin untersucht werden.

Die Arbeit ist folgendermassen aufgebaut:

Kapitel 2 beschreibt die verschiedenen Regelwerke, die unter der Schirmherrschaft des Qualitätsmanagements in den letzten Jahren entwickelt wurden, um der zunehmenden Komplexität von Software-Entwicklungsprojekten Herr zu werden. Neuere Ansätze, die aus dem Gebiet des Usability Engineerings hervorgegangen sind, verdeutlichen, dass dem Problem nicht allein mit technikzentrierten Massnahmen begegnet werden kann. Neben allen Bemühungen auf Prozessebene werden auch die Aspekte behandelt, die im Rahmen der Produktprüfung den Erfolg der Bemühungen zeigen können.

Neben den konkreten Inhalten der verschiedenen Normen und Prozessmodelle geht es in Kapitel 2 vor allem darum, die grundlegenden Ansätze zu untersuchen, die für die Präsentation, Analyse und Optimierung von Prozessen herangezogen werden.

Kapitel 3 behandelt die wichtigsten Usability Engineering Prozesse, die in den letzten Jahren hauptsächlich in den USA entworfen wurden. Neben grundlegenden Gemeinsamkeiten, wie beispielsweise frühe Benutzerinvolvierung und iterative Entwicklung werden in diesem Kapitel auch die Unterschiede der Usability Engineering Prozesse aufgezeigt, die vor allem hinsichtlich einer Integration von Software-Engineering und Usability Engineering folgenreich sind. Probleme, die mit der praktischen Umsetzung der Prozesse zu erwarten sind, werden in Kapitel 3.2 diskutiert, zusammen mit den grundlegenden Überlegungen zu deren Lösung.

Der Blickwinkel dieser Arbeit reicht über die Prozess- oder Methodenebene hinaus. Überlegungen, was es denn so schwierig macht, die naheliegenden Herangehensweisen des Usability Engineerings zur Lösung zahlreicher Probleme des Software-Engineerings heranzuziehen, haben dazu geführt, Benutzerzentrierung als Innovation zu betrachten. Kapitel 4 soll einen kleinen Einblick in die Innovationstheorie geben, aus der sich einige zentrale Aspekte ableiten lassen, die im Hinblick auf die Förderung von Benutzerzentrierung relevant sind.

Kapitel 5 stellt das Kernstück der Arbeit dar. Hier wird als Fazit der vorhergehenden Kapitel ein Modell von einer optimal benutzerzentrierten Organisation entwickelt. Anhand von sieben Dimensionen (Prozessmodell, Methoden und Tools, Organisationsstruktur, Managementaktivitäten und Kultur, Qualifikation der Mitarbeiter, Schulungswesen und Wissensmanagement sowie Ressourcen) werden Kriterien definiert, die eine Organisation erfüllen müsste, die sich wirklich benutzerzentriert nennen darf. Das vorgestellte Modell ist angelehnt

an ein Reifegradmodell, wie man es auch aus dem Software-Engineering kennt ([Capability Maturity](#) Model). In Kapitel 5.2 wird ein Ratingverfahren vorgestellt, das eine differenzierte Analyse des Reifegrads einer Organisation erlauben soll. Ziel des gesamten Modells und des Ratingverfahrens ist es, Hinweise für Optimierungsmassnahmen auf dem Weg zu einer benutzerzentrierten Organisation zu erhalten.

In Kapitel 6 wird das entwickelte Modell der benutzerzentrierten Organisation auf UBS Schweiz angewendet. Zunächst werden Besonderheiten der Software-Entwicklung im Umfeld UBS Schweiz und die Stellung des Usability Engineerings in diesem Gefüge beschrieben (Kapitel 6.1). In Kapitel 6.2 wird unter Anwendung des Modells eine Analyse des Stands der Benutzerzentrierung bei UBS Schweiz durchgeführt.

Auf der Basis der vorhergegangenen Analyse werden in Kapitel 7 konkrete Massnahmen zur Verbesserung des Stands der Benutzerzentrierung von UBS Schweiz abgeleitet. Die Massnahmen werden unterteilt in strategische Massnahmen (Kapitel 7.1) und technologische Infrastrukturmassnahmen (Kapitel 7.2). Im Gegensatz zu den strategischen Massnahmen, deren Durchführung grösstenteils auch heute schon realistisch erscheint, stellen die Infrastrukturmassnahmen im Falle von UBS Schweiz hauptsächlich Zukunftsmusik dar.

Mit einer Reflexion über Erfahrungen, die ich während der Erstellung dieser Arbeit gesammelt habe, über erste Ideen und Pläne, über Bord geworfene Ansprüche bis hin zu Erkenntnissen, die heute klarer sind als noch vor einem halben Jahr, endet diese Arbeit (Kapitel 8).

2 Software-Entwicklungsprozesse unter der Lupe

Inzwischen gibt es eine ganze Landschaft an Normen, Prozessmodellen und Prüfmethode im Bereich der Software-Entwicklung, die alle dazu beitragen sollen, die Probleme des Software-Engineerings besser beherrschbar zu machen und die Qualität der zu entwickelnden Produkte zu steigern. Zunehmend wird auch die Usability des Endprodukts als Erfolgsfaktor gewichtet. Dies führte zur Entwicklung von benutzerzentrierten Prozessen und zu einem erweiterten Betrachtungsrahmen der Software-Entwicklung.

Bevor das Rad komplett neu erfunden wird, ist es sinnvoll zu evaluieren, was vorhandene Ansätze zur Verbesserung der Softwarequalität bieten. Im folgenden sollen daher die existierenden Normen und Prozessmodelle kurz umrissen und die für diese Arbeit interessant erscheinenden Aspekte herausgestellt werden.

2.1 Qualitätsmanagement nach ISO 9001 und ISO 9000-3

Mit der Idee, dass Produktqualität das Resultat eines guten Prozesses ist, haben sich die Bemühungen um die Verbesserung von Produkten aller Art in der Vergangenheit mit Schwerpunkt auf der Prozessebene abgespielt. Die ISO 9000er Serie der International Standardization Organization aus dem Jahr 1985 ist sozusagen die Urmutter des standardisierten Qualitätsmanagements (QM). Sie wurde auch vielfach auf die Entwicklung von Softwareprodukten angewendet. Eine Interpretation für den Software-Entwicklungsbereich ist jedoch nicht ganz einfach, da mit der Normenreihe in erster Linie klassische Fertigungsbetriebe angesprochen werden. Aus diesem Grund wurde 1991 der Leitfaden ISO 9000-3 entwickelt, um die Anwendung von ISO 9001 zu erleichtern. (ISO 1994), (DIN EN ISO 1998)

Der Erfolg der ISO 9000er Reihe im Bereich der Software-Entwicklung ist eher gering. Dies liegt nicht zuletzt daran, dass diese Form der Qualitätssicherung nur das eingesetzte Verfahren prüft und zertifiziert, jedoch keinerlei Anforderungen an das zu erstellende Produkt formuliert (Balzert 1998). Dies führte dazu, dass im Software-Entwicklungsbereich QM-Systeme nach ISO 9001 „kaum zu einer verbesserten Einhaltung von Zeit- und Kostenplänen in den Projekten geführt haben; von einer erzielten Verbesserung der Produktqualität ist nichts bekannt.“ (zitiert nach Deutsche Akkreditierungsstelle Technik e.V. 2001a, S. 4, Stelzer 2000)

Bewertung im Hinblick auf diese Arbeit

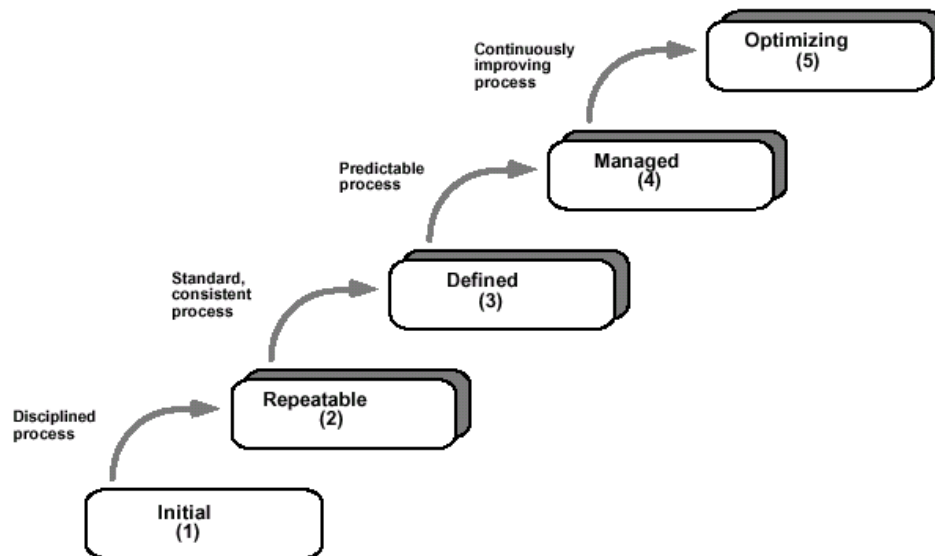
Aus den genannten Gründen wird die ISO 9001 bzw. ISO 9000-3 nicht weiter betrachtet.

2.2 Die Prozessqualität

2.2.1 Das Capability Maturity Model

Um ein Mass für die Leistungsfähigkeit von Software-Lieferanten zu erhalten, wurde bereits 1987 vom Software Engineering Institute (SEI) der Carnegie Mellon University ein Fragebogen erstellt, der später zu einem Referenzmodell ausgebaut wurde, das den Namen *Capability Maturity Model* (CMM) erhielt. Kern des CMM ist es, mit Hilfe sogenannter Assessments die Reifegradstufe eines Software-Entwicklungsunternehmens (oder eines Unternehmensbereiches) zu ermitteln. Im Rahmen der Prozessbewertung erhält man zudem Hinweise für die inkrementelle Verbesserung der Prozessqualität.

Die Grundidee von CMM ist, dass umso eher Projektziele richtig formuliert und auch erreicht werden können (*Capability*), je strukturierter und definierter der Prozess selbst und je ausgefeilter seine Steuerung und Dokumentation sind. Laut Untersuchungen des SEI an 3500 Softwareprojekten in den USA (Mai 1998) befinden sich 58% der Projekte im Initial Level, 24% im Repeatable Level, 12% im Defined Level, 2% im Managed Level und 1% im Optimized Level. Vergleiche mit Studien von 1991 zeigen, dass sich die Prozessqualität deutlich verbessert hat (Schönwälder 2002, S. 14). Entwicklungsprozesse lassen sich mit Hilfe von CMM-Reifegradstufen (*Maturity Levels*) qualitativ beurteilen (vgl. hierzu Abbildung 1). Jede Reifegradstufe zeichnet sich durch die Berücksichtigung bestimmter Kernprozesse (Key Process Areas) aus. Für jeden Kernprozess sind wiederum Ziele formuliert, die im Entwicklungsprozess erfüllt werden müssen. Spezifische Fragestellungen des Usability Engineerings, die z.B. in den beiden Kernprozessen ‚Requirements Management‘ (Reifegradstufe 2) und ‚Software Product Engineering‘ (Reifegradstufe 3) angesiedelt sein könnten, werden nicht behandelt.



(Paulk et al. 1997)

Abbildung 1: Die 5 Reifegradstufen des CMM

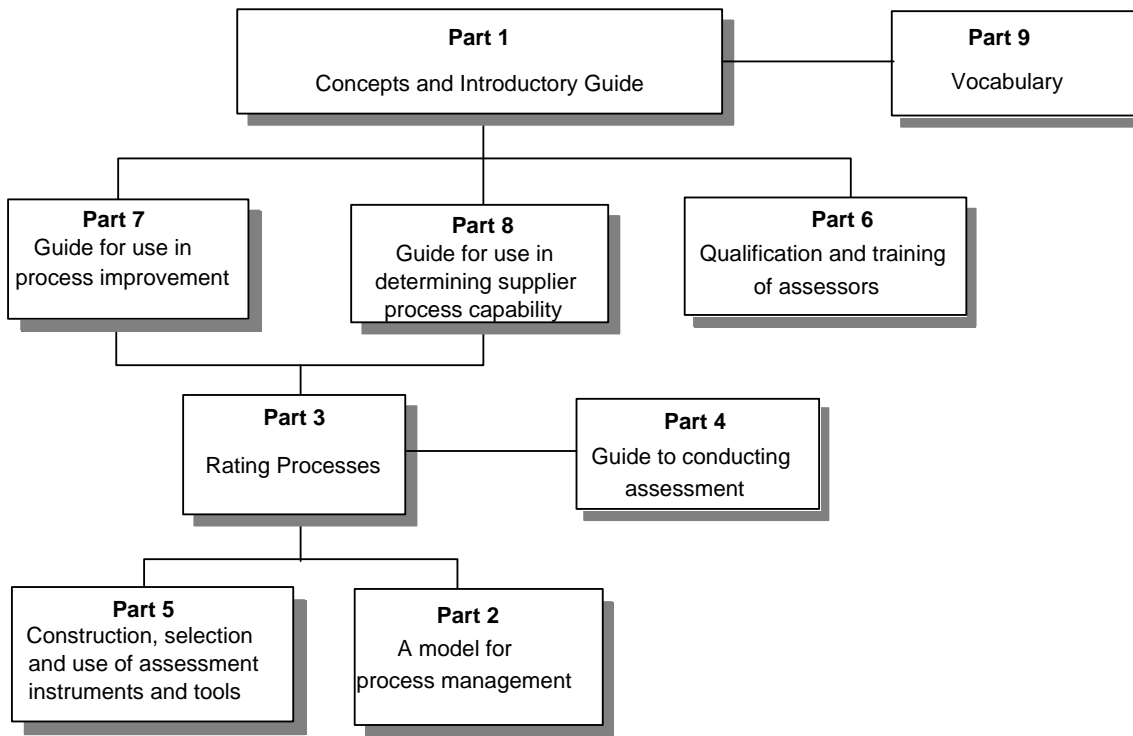
Bewertung im Hinblick auf diese Arbeit

CMM ermöglicht einen strukturierten Entwicklungsprozess, der dazu tauglich ist, den Zeit- und Kostenrahmen von Software-Entwicklungsprojekten besser zu erfüllen. Weitergehende Zielsetzungen, wie z.B. die Usability eines Produktes, werden innerhalb von CMM nur in dem Rahmen abgedeckt, wie sie im herkömmlichen Requirements-Engineering auch anzutreffen sind. Der *Capability*-Massstab ist jedoch interessant, denn er kann entweder zur Qualitätsbewertung eines Unternehmens herangezogen werden oder aber, um Massnahmen zur Qualitätsverbesserung zu planen. Dies gilt allgemein für jede Art der im folgenden vorgestellten Reifegradmodelle.

2.2.2 ISO 15504 (SPICE)

1998 wurde die ISO-Norm 15504, auch als SPICE (**S**oftware **P**rocess **I**mprovement and **C**apability **D**etermination) bekannt, verabschiedet. ISO 15504 hat zum Ziel, „einen umfassenden, ordnenden Rahmen zur Bewertung und Verbesserung von Software-Prozessen“ zur Verfügung zu stellen (Balzert 1998, S. 377), (vgl. Abbildung 2). Die Ideen von SPICE sind teilweise der Norm ISO 12207 entlehnt, in der ebenfalls verschiedene Prozessdimensionen definiert sind, die in Form spezifischer Aktivitäten und Aufgaben operationalisiert werden. ISO 12207 beschränkt sich jedoch hauptsächlich auf die Regelungen eines Vertragsverhältnisses im Rahmen von Software-Entwicklungsprozessen und wird daher an dieser Stelle nicht weiter betrachtet. (ISO/IEC 1995a)

Mit SPICE soll die Selbstbewertung von Unternehmen hinsichtlich des eigenen Software-Entwicklungsprozesses ermöglicht werden, wobei sowohl die Vollständigkeit des Prozesses („Prozess-Dimension“) als auch die Leistungsfähigkeit des Prozesses („Reifegrad-Dimension“) bewertet werden. Im Gegensatz zu CMM werden bei SPICE nicht Unternehmen oder Organisationseinheiten evaluiert, sondern ein einzelner Prozess. Die Prozess-Assessments erlauben einen guten Einblick in den Prozess, wodurch das Verbesserungspotential ersichtlich wird.

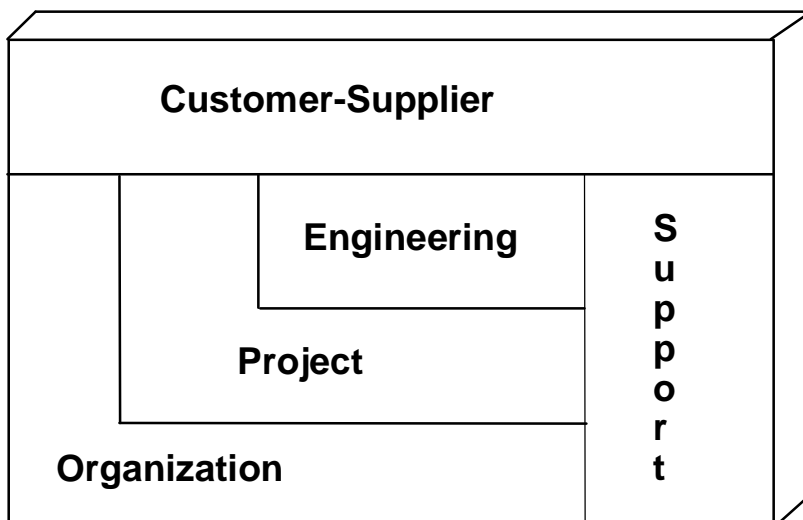


(ISO/IEC 1995b)

Abbildung 2: Bestandteile von SPICE

In Teil 2 von SPICE wird ein generisches Prozessmodell für die Software-Entwicklung vorgestellt. Es werden vergleichbar mit CMM 5 Prozessreifegrade definiert, wobei insgesamt 5 Prozesskategorien bewertet werden, die den Rahmen des eigentlichen Entwicklungsprozesses z.T. deutlich überschreiten. Die Prozesskategorien beziehen sich auf den operationellen Bereich (Customer – Supplier) und auf die Produktion (Engineering, Project, Support), die den eigentlichen Entwicklungsprozess abbildet.

Für jede dieser Kategorien werden Prozesse definiert, die über Aktivitäten (sogenannte base practices) realisiert werden. Die Kategorien gruppieren die gesamten Projektaktivitäten thematisch und verzichten auf eine zeitliche Zuordnung innerhalb des Prozesses.



(ISO/IEC 1995c, Annex C, S. 2)

Abbildung 3: Prozesskategorien in SPICE / ISO 15504

Teil 7 von SPICE beschreibt Grundlagen, generelle Prinzipien und Vorgehensweisen zur Prozessverbesserung. Es wird darauf hingewiesen, dass erfolgreiche Veränderungen immer von der Spitze einer Organi-

sation getragen werden müssen. Das obere Management muss hierbei die Führerschaft übernehmen und die entsprechenden Ressourcen bereitstellen. Ausserdem müssen die Veränderungen von den Mitarbeiterinnen mitgetragen werden, die hierzu das notwendige Verständnis über den vorhandenen Prozess und die angestrebten Veränderungsziele besitzen müssen.

Zusammenfassend beschreibt SPICE folgende grundlegenden Prinzipien für die Erreichung von Prozessverbesserungen: (ISO/IEC 1995d)

- Verbesserungen basieren auf Prozessassessments und Effektivitätsmessungen;
- Verbesserungsaktivitäten werden identifiziert und priorisiert;
- Verbesserungen sind ein kontinuierlicher Prozess mit identifizierten, innerhalb der Organisation akzeptierten Zielen und werden über ein Programm realisiert;
- Eine Risikoabschätzung bezüglich der momentanen Situation und dem Fall des Scheiterns der Verbesserungsinitiative ist Bestandteil des Verbesserungsprozesses.

Bewertung im Hinblick auf diese Arbeit:

SPICE zielt auf Selbstbeurteilung auf Projektebene ab. Da Software-Entwicklungsprojekte auch innerhalb einer Organisation sehr unterschiedlich sind, erscheint diese differenzierte Vorgehensweise angemessen.

Anstelle eines konkreten Prozessmodells werden Aktivitäten zu fünf verschiedenen Prozesskategorien definiert, die als Dimensionen einer erfolgreichen Entwicklungsorganisation (nicht nur eines Entwicklungsprozesses) zu sehen sind. Eine Erweiterung dieser Prozesskategorien ist möglich (es gibt ein Template und Vorgaben dafür).

Die Definition von (Prozess-)Kategorien (vgl. Abbildung 3) ist ein interessanter Ansatz zur Bündelung von Aktivitäten, ersetzt jedoch vermutlich nicht die Beschreibung dieser Aktivitäten in einem irgendwie gearteten zeitlichen Ablauf.

Der Rating Prozess von SPICE bewertet zum einen die Existenz (durchgeführt / nicht durchgeführt) einer Prozessaktivität (*base practices*) als auch deren **adäquate** Durchführung. Ebenso wird die adäquate Durchführung der Aktivitäten bewertet, die für die Implementation eines Prozesses relevant sind (*generic practices*). Diese Art des Ratings ist nachahmenswert, da nicht unbedingt auf Vollständigkeit, sondern vor allem auf Adäquatheit einzelner Aktivitäten Wert gelegt wird. Das Rating ermöglicht anschliessend nicht nur Prozessvergleiche, sondern auch das Herausarbeiten systematischer Verbesserungsoptionen.

Obwohl SPICE im Hinblick auf Usability wenig bietet, ist auch die Vorgehensweise zur Prozessoptimierung, wie sie in Teil 7 beschrieben wird, so generisch, dass sie auf die Optimierung von Usability Engineering Prozessen anwendbar erscheint.

2.2.3 Benutzerzentrierung: ISO 13407, DATech Prüfbaustein und Usability Maturity Model

ISO 13407 definiert die Prinzipien benutzerzentrierter Software-Entwicklungsprozesse:

- a) Aktive Einbeziehung von Benutzern und genaues Verständnis der Benutzerin und ihrer Aufgaben;
- b) Angemessene Aufteilung der Funktionen zwischen Benutzer und Technologie;
- c) Iterativer Designprozess;
- d) Multidisziplinäre Designteams.

Eine korrekte Durchführung eines solchen Prozesses bedarf einer umfassenden Planung aller benutzerzentrierten Aktivitäten. Folgende Prozessschritte müssen abgedeckt sein:

1. Verstehen und Spezifizieren des Benutzungskontextes;
2. Spezifikation der Benutzeranforderungen und der Organisationellen Anforderungen;
3. Produktion von Designlösungen;
4. Evaluation der Designlösungen anhand der spezifizierten Anforderungen.

(DIN EN ISO 1999)

Auf der Basis der ISO 13407 wurde mit Version 1.1 im Jahr 2001 der **DATech-Prüfbaustein Usability Engineering Prozess** der Deutschen Akkreditierungsstelle Technik e.V. veröffentlicht. Der Prüfbaustein soll dazu dienen, den Entwicklungsprozess von Herstellerorganisationen zu bewerten. Das Dokument entwirft Gütekriterien, die als Grundlage für die Entwicklung von *base practices* zu sehen sind, die von Software-Herstellern im Sinne eines benutzerzentrierten Entwicklungsprozesses durchgeführt werden sollten und als Ergänzung zu vorhandenen Aktivitäten der Software-Entwicklungsprozesse zu betrachten sind. Wie diesen herkömmlichen Modellen liegt auch hier der Gedanke zugrunde, dass ein adäquater Usability Engineering Prozess zu einem gebrauchstauglichen Produkt führt.

Der DATech-Prüfbaustein deckt 2 Prozesskategorien ab (vergleichbar mit den Kategorien von ISO 15504): Die ‚Anwender-/ Benutzerorientierung‘, mit Parametern aus 6 der insgesamt 15 Beurteilungsdimensionen, und die Kategorie ‚Organisatorische und fachliche Steuerung‘ mit Parametern aus 10 Beurteilungsdimensionen. In den Beurteilungsdimensionen werden bestimmte Aktivitäten definiert, die alle bereits auf der ersten Reifestufe abgedeckt werden müssen. Der Anspruch an die Durchführung der Aktivitäten steigt jedoch mit jeder Reifestufe.

Es werden nur noch 3 Reifestufen unterschieden:

- Stufe 1: Anfangsstadium des UE-Prozesses → Usability Engineering Aktivitäten werden zwar durchgeführt, jedoch nicht von Anfang an, nicht systematisch und ohne Integration von Experten ins Projektteam.
- Stufe 2: Reproduzierbare Effektivität des UE-Prozesses → der Projekterfolg hinsichtlich Usability ist reproduzierbar, Usability Anforderungen werden validiert, Fachleute sind ins Projektteam integriert, Benutzerinnen werden systematisch einbezogen.
- Stufe 3: Sich kontinuierlich verbessernder UE-Prozess → Mechanismen zur systematischen Prozessoptimierung sind etabliert. Usability Engineering ist innerhalb der Herstellerorganisation als Querschnittsaufgabe verankert. Relevantes Wissen im Zusammenhang mit Usability von Produkten sowohl in der Entwicklung als auch in der Anwendung wird systematisch erfasst und verwaltet.

Das Erreichen einer Stufe verlangt die Erfüllung aller Anforderungen der angestrebten Stufe. Die Erfüllung der darunterliegenden Stufen ist automatisch gegeben, da sich die Ansprüche kontinuierlich von Stufe zu Stufe steigern. Eine Konformität mit ISO 13407 ist ab Stufe 2 gewährleistet.

Prüfobjekt beim DATech Prüfbaustein ist ein Projekt, bei dem Usability Engineering stattgefunden hat, also nicht die gesamte Organisation. (Deutsche Akkreditierungsstelle Technik e.V. 2001a)

Usability Maturity Model: Processes (TRUMP Version)

Ziel von UMM ist es, zu messen, wie gut eine Organisation den Anteil der Software-Entwicklung abhandelt, der mit Benutzerzentrierung zu tun hat. UMM definiert ein Prozessmodell, bestehend aus 7 Human Centred Design Processes (HCD), die in erster Linie auf ISO 13407 basieren (HCD 1 und 7 ergänzen die ISO 13407). Für jeden HCD Prozess werden *base practices* definiert, die beschreiben, was im Laufe eines Software-Engineering-Lifecycles durchzuführen ist, um die Benutzer eines Systems in den Prozess einzubeziehen (im Prinzip ein klassisches Usability Engineering Prozessmodell). Die Formulierung der Basisaktivitäten ist ergebnisorientiert, Methoden zur Durchführung werden nicht vorgeschrieben und auch nicht empfohlen.

Die Bewertung des Prozesses findet in Form von *Capability* Levels statt (angelehnt an ISO 15504), wobei zum einen definierte Prozessattribute bewertet werden, zum anderen aber auch Managementaktivitäten, die entscheidend sind, für die notwendige Kontrolle über den Entwicklungsprozess. Es gibt 5 *Capability*-Levels (und den Level 0 für einen nicht implementierten oder unbrauchbaren Prozess), die aufeinander aufbauen. Auf jedem *Capability* Level werden zusätzliche *base practices* gefordert. Ein Level kann nur dann erreicht werden, wenn auch alle Anforderungen an darunterliegende Levels vollständig erfüllt sind. Auch das UMM Prozessmodell kann sowohl dazu verwendet werden, einen benutzerzentrierten Prozess zu definieren oder zu verbessern als auch, um einen Entwicklungsprozess hinsichtlich seiner Benutzerzentrierung zu bewerten. (Earthy 1999)

Bewertung im Hinblick auf diese Arbeit

ISO 13407 ist ein erster Schritt zur Integration von Software-Engineering und Usability Engineering, der im DATech Prüfbaustein komplett aufgegriffen und erweitert wird. Die beiden Prozesskategorien ‚Anwender-/ Benutzerorientierung‘ und ‚Organisatorische und fachliche Steuerung‘ mit ihren jeweiligen Parametern und die Bewertungsdimensionen des DATech-Prüfbausteins liefern zahlreiche brauchbare Aspekte, die nach Möglichkeit übernommen werden sollen. Ein 3-stufiges Bewertungsschema scheint ausreichend, allerdings ist hier der DATech-Prüfbaustein fragwürdig, da er auf Vollständigkeit anstatt auf Adäquatheit abstellt. Diesbezüglich bleibt zu ermitteln, ob der SPICE Ansatz nicht sinnvoller wäre.

Am UMM Prozessmodell ist die Definition von HCD Prozessen und deren zeitliche Darstellung ein Problem, da die Iterativität der Prozesse nicht visualisiert wird. Allerdings liefert das UMM einen konkreten Prozess, der als solcher auch leicht verständlich und kommunizierbar ist. Spannend am UMM Ansatz ist die Betonung der Wichtigkeit von Managementprozessen.

Konkrete Methoden sind in keinem der drei besprochenen Ansätze vorgeschrieben, was zur Erhaltung der Freiheitsgrade der Anwenderinnen zwar sinnvoll ist. Ohne Hinweise auf Hilfsmittel und best practices ist jedoch zu befürchten, dass die Durchführung der Aktivitäten nicht von der gewünschten Qualität ist.

2.2.4 Das Organisationelle Umfeld: Usability Maturity Model: Human Centredness Scale

Als eine wichtige Facette erfolgreichen Usability Engineerings neben der eigentlichen Prozessdurchführung ist das Mass und die Art und Weise zu sehen, wie die Organisation selbst die Benutzerzentrierung fördert und unterstützt. Das Dokument ‚UMM: Human Centredness Scale‘ definiert ein einfaches Modell für eine Organisation, die benutzerzentriert Software entwickelt. Gleichzeitig werden Methoden erklärt, mit deren Hilfe der Reifegrad einer Organisation im Hinblick auf Benutzerzentrierung ermittelt werden kann. Organisationen können auf einer Reifeskala von A – E oder X (unbekannt) eingestuft werden. Zur Bewertung werden Management Praktiken herangezogen. Es wird nicht die persönliche Einstellung gegenüber den Endbenutzerinnen einer Organisation bewertet, sondern das Mass, in dem eine Organisation Praktiken implementiert hat, die die Prozesse benutzerzentriert machen. Für das Rating gibt es eine abgestufte Bewertung (‚largely‘ oder ‚fully‘). Die beschriebenen Attribute beziehen sich auf den Prozess selbst (z.B. Benutzerfokus, Benutzerinvolvierung) oder auf die Rahmenbedingungen der Durchführung des Prozesses (z.B. Qualifikation der Durchführenden von benutzerzentrierten Vorgehen oder angewandte Techniken). Es finden sich aber auch übergeordnete Attribute wie z.B. Bewusstsein bezüglich des Usabilityproblems und Wahrnehmung des Themas Nutzungsqualität.

Der Level an Benutzerzentriertheit einer Organisation gibt Auskunft darüber, wie man innerhalb der Organisation Belange der Benutzerzentrierung am besten kommuniziert. Die Idee dahinter ist, dass eine Organisation mit einem tiefen Rating nicht in der Lage ist, die Prozesse erfolgreich durchzuführen, die zur Erreichung eines hohen Levels notwendig sind. Allerdings wird sie die Anforderungen des nächsthöheren Levels verstehen und einsehen, was die Vorteile davon sind. Dies impliziert, dass Prozessverbesserung zur Erreichung eines höheren Levels nur kontinuierlich und nicht ad hoc möglich ist. (Earthy 1998)

Bewertung im Hinblick auf diese Arbeit

UMM bewertet den Prozess und die Einstellung der Organisation gegenüber Benutzerzentrierung getrennt voneinander. Dies ist insofern interessant, da die verschiedenen Bereiche durchaus auf einem unterschiedlichen Level bezüglich Benutzerzentriertheit sein können, auch wenn letztlich alle Bereiche entscheidend dafür sind, ob innerhalb der Organisation erfolgreich benutzerzentriert entwickelt wird (jetzt und in der Zukunft). Durch die konkrete Aufspaltung ist es möglich, jedem einzelnen Bereich die notwendige Aufmerksamkeit zu schenken und die Leistungsfähigkeit und *Capability* getrennt zu betrachten um daraus Verbesserungsmöglichkeiten abzuleiten. Ob jedoch die Differenzierung in Prozess und Organisation ausreicht und in dieser Art sinnvoll ist, ist fraglich. Gegebenenfalls müsste eine weitere Differenzierung gesucht werden. Zudem fällt auf, dass sich die Aktivitäten des Dokumentes ‚Usability Maturity Model: Human Centredness Scale‘ über weite Strecken immer noch auf der Ebene des Entwicklungsprozesses selbst bewegen (z.B. Aspekte wie die systematische Sammlung und Einbeziehung von Benutzer-Feedback usw.). Bezüglich der Attitüde der Organisation und deren Rahmenbedingungen mit Einfluss auf die Benutzerzentriertheit sollte diese Idee noch erweitert werden.

2.3 Die Produktqualität

Obwohl es einen kausalen Zusammenhang zwischen Prozess und Produktqualität gibt, kommt man nicht umhin, auch die Qualität des Produktes zu ermitteln. Im Falle der Usability darf zudem der Aspekt nicht vernachlässigt werden, dass Usability nicht nur die Eigenschaft eines Produktes ist, sondern direkt an den Nutzungskontext gekoppelt werden muss.

Für die Förderung von Benutzerzentrierung innerhalb einer Entwicklungsorganisation ist vor allem wichtig, dass zur Prüfung der Produktqualität auch eine Prüfung der Usability gehört. Die eigentliche Prüfung selbst ist nicht Thema dieser Arbeit, die verschiedenen Ansätze innerhalb der vorhandenen Regelwerke sollen hier jedoch der Vollständigkeit halber kurz erwähnt werden.

2.3.1 Allgemeine Produktqualität: ISO 9126-1

In der ISO Norm 9126, Teil 1 (erstmalig erschienen im Jahr 1991 und nun in einer überarbeiteten Version im Jahr 2001) wird ein allgemeines Qualitätsmodell für Softwareprodukte verwendet, das auch den Usability Begriff umfasst. Der in der ISO 9126 – 1 definierte Qualitätsbegriff greift neben den eigentlichen Produktattributen (wie beispielsweise Funktionalität, Zuverlässigkeit, Usability usw.) erstmalig auch den Bereich der Nutzungsqualität auf (Quality of use). Anders als die anderen qualitativen Attribute ist dieser Aspekt jedoch nicht ohne den jeweiligen Nutzungskontext zu bewerten. (ISO/IEC 2001)

2.3.2 Die Usability des Produktes: ISO 9241 (Teil 10 – 17) und ISO 12119

Als grundlegende Norm der Software-Ergonomie kann die ISO 9241 bezeichnet werden. Die Teile 10 – 17 konzentrieren sich ausschliesslich auf die Anforderungen der Mensch-Computer Schnittstelle. Die Usability eines Produktes wird durch folgende Merkmale definiert (ISO 9241 – 11):

1. Effektivität: vollständige und korrekte Erreichung der Nutzungsziele im vorgesehenen Nutzungskontext;
2. Effizienz: der im Verhältnis zur Genauigkeit und Vollständigkeit eingesetzte Aufwand, mit dem die Benutzerinnen ein bestimmtes Ziel erreichen;
3. Benutzerzufriedenheit: Beeinträchtigungsfreiheit der Nutzung und positive Einstellungen der Benutzer zur Produktnutzung.

ISO 12119 und Teile der ISO 9241 operationalisieren diese Eigenschaften:

- ISO 12119 beschreibt die grundlegenden Anforderungen an jede Dialogsoftware: Dokumentation, Verständlichkeit, Funktionalität, Stabilität und Zuverlässigkeit. Diese sind die Grundlage für die Effektivität der Nutzung.
- ISO 9241 – 10 formuliert die Dialogprinzipien: Aufgabenangemessenheit, Erwartungskonformität, Steuerbarkeit, Fehlertoleranz und Selbstbeschreibungsfähigkeit, die massgeblich die Effizienz eines Produktes beeinflussen;
- ISO 9241 – 11 beschreibt das Vorgehen zur Spezifikation und Evaluation der Gebrauchstauglichkeit im Rahmen der Produktentwicklung. Hierzu definiert die Norm die erforderlichen Informationen, die zur Bestimmung der Gebrauchstauglichkeit notwendig sind: (1) Ziele, (2) Nutzungskontext (Benutzer, deren Ziele und Aufgaben sowie die Anwendungsumgebung), (3) angestrebte quantitative Werte der Gebrauchstauglichkeit im vorgesehenen Nutzungskontext. Damit legt ISO 9241 – 11 nicht nur die Basis für die Spezifikation und Entwicklung, sondern auch für die Prüfung der Gebrauchstauglichkeit.

Zusätzlich zu diesen Produktmerkmalen definiert ISO 9241 – 2 noch die Anforderungen an den Nutzungskontext, die unabhängig vom Produkt die Zufriedenstellung der Benutzerinnen beeinflussen. Teil 12 und 13 (Informationsdarstellung und Benutzerführung) der ISO 9241 geben Hinweise zur praktischen Umsetzung der Prinzipien aus Teil 10. Die Teile 14 – 17 beziehen sich differenziert auf einzelne Dialogtechniken bzw. –elemente und stellen damit konkrete Gestaltungshilfen dar, um die Einhaltung der Dialogprinzipien zu ermöglichen. (ISO/IEC 1994), (ISO 1992), (DIN EN ISO 1996-1999)

2.3.3 Die Produktprüfung: DATech-Prüfhandbuch Gebrauchstauglichkeit

Mit dem Ziel, Prüflabors zu akkreditieren, ist im Jahr 2001 im Zusammenhang mit dem DATech-Prüfbaustein das DATech-Prüfhandbuch Gebrauchstauglichkeit entstanden. Das Prüfhandbuch dient einerseits zur Durchführung von Software-Konformitätsprüfungen mit den Vorgaben von ISO 9241 – 10, zum anderen kann es verwendet werden, um Anwendungsvereinbarungen für Entwicklungsorganisationen und Anwenderinnen zu erstellen.

Das DATech-Prüfhandbuch Gebrauchstauglichkeit beschreibt die Vorgehensweise, wie auf der Basis von Benutzer-, Aufgaben- und Kontextanalysen qualitative und quantitative Usability Ziele als Abnahmekriterien für die Produktprüfung erstellt werden. Eine sinnvolle Prüfung setzt eine Priorisierung der Ziele und möglichen Testfälle (z.B. Use Cases) voraus.¹ Werden im Rahmen der Prüfung Abweichungen zwischen Zielen und Ergebnissen ersichtlich, muss ermittelt werden, welche Wirkungen diese Abweichungen auf die tatsächliche Nutzungsqualität des Produktes haben (Erhärtungsprüfung). (Deutsche Akkreditierungsstelle Technik e.V. 2001b)

¹ In einem benutzerzentrierten Prozess sollten diese Vorarbeiten selbstverständlich längst vor der abschliessenden Produktprüfung erledigt sein, da sie die Basis für das Design eines gebrauchstauglichen Produktes darstellen.

2.4 Zusammenfassung

Die hier betrachtete Normen- und Prozesslandschaft macht deutlich, wie umfangreich die Bemühungen sind, die zunehmend komplexen Software-Entwicklungsprozesse besser beherrschbar zu machen.

Im Rahmen eines umfassenden Qualitätsmanagements muss sowohl die Prozess- als auch die Produktebene betrachtet werden. Dies bedingt die Definition geeigneter Prozessmodelle. Auf Produktebene muss die Qualität anhand vorher definierter Qualitätsmerkmale geprüft werden. Im Zusammenhang mit dem Qualitätsmerkmal Usability ist zudem der Nutzungskontext zu definieren und bei der Prüfung zu berücksichtigen. Dass Usability und Usability Engineering zumindest in der Normenwelt kein Schattendasein führen, zeigt die schiere Fülle der diesbezüglich vorhandenen Normen, die gleichermassen Prozess- und Produktqualität abdecken.

Kontinuierliche Prozessverbesserung ist eine Kernanforderung des Qualitätsmanagements. Eine naheliegende Antwort auf diese Anforderung sind die vorgestellten Reifegradmodelle (CMM, SPICE und UMM). Interessanterweise werden in einigen der vorgestellten Modelle auch Dimensionen behandelt, die ausserhalb des eigentlichen Prozesses angesiedelt sind. Diese organisationellen Dimensionen werden als Grundlage für ein wirksames Qualitätsmanagement betrachtet. In Bezug auf die Usability Engineering Prozesse bekommt die Attitüde der Organisation eine gewichtige Bedeutung, da sie ausschlaggebend dafür ist, welche Prozessoptimierungen überhaupt erfolgsversprechend sind. Abbildung 4 zeigt die vorgestellten Normen- und Prozessmodelle in ihrem Zusammenhang:

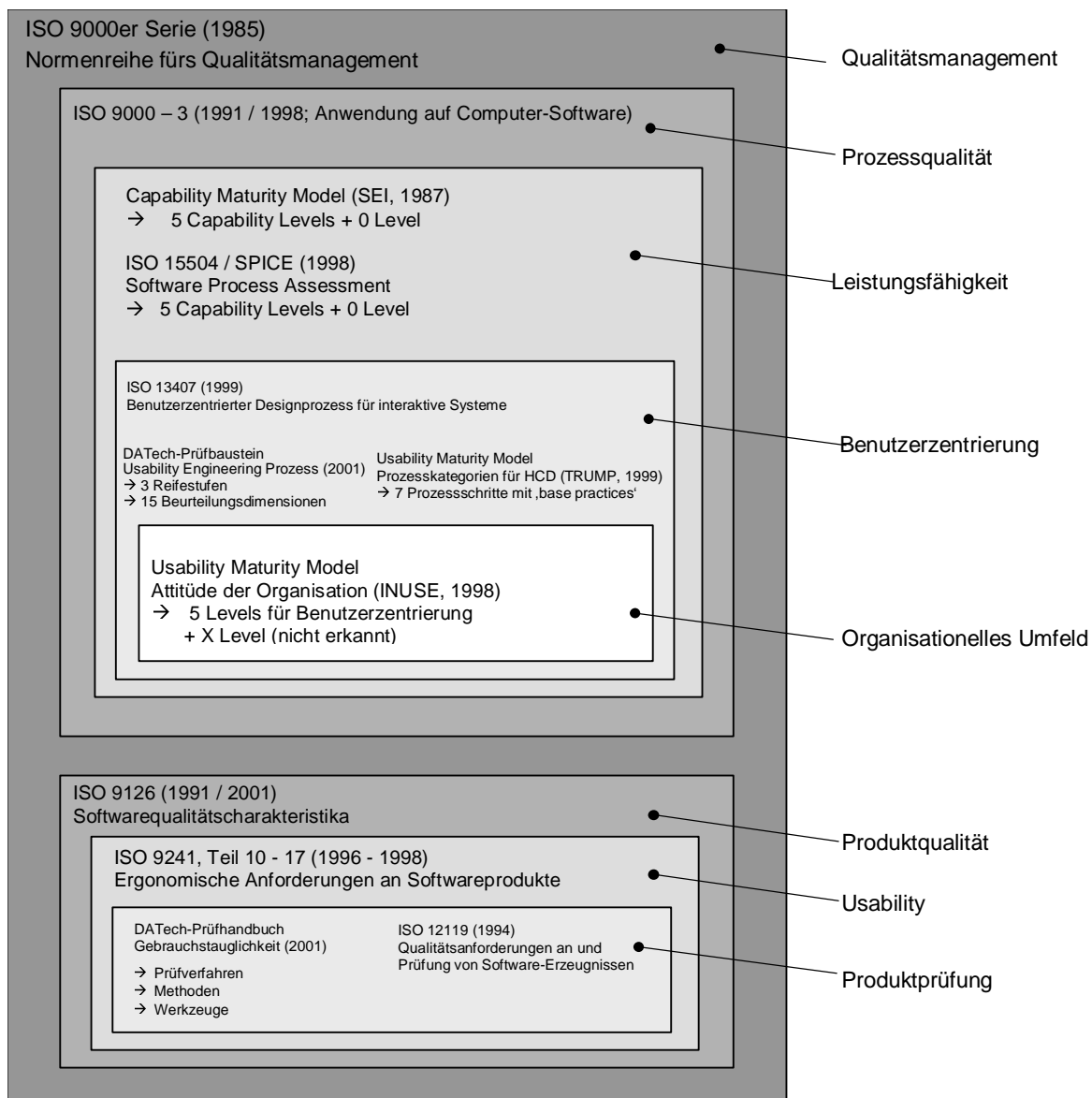


Abbildung 4: Normen und Prozessmodelle in der Software-Entwicklung

Wie die Normenlandschaft zeigt, besteht nach wie vor eine gewisse Trennung zwischen Software-Engineering und Usability Engineering Modellen. Die Definition der Produktqualität stellt die Schnittstelle zwischen diesen Welten dar. Usability ist bezüglich der Qualität von Software-Produkten ein zentrales Merkmal, das sich heute auch in den software-technisch orientierten Normen, wie bspw. der neuen ISO 9126 wiederfindet. Es ist davon auszugehen, dass zukünftig eine stärkere Integration von Usability Engineering und Software-Engineering Ansätzen stattfinden wird.

3 Usability Engineering Prozesse in Theorie und Praxis

Usability Engineering sollte idealerweise nicht in Form von Einzelmethoden, sondern in einem übergeordneten «Lifecycle» durchgeführt werden, dessen Aktivitäten weit vor dem eigentlichen User Interface Design beginnen und auch die Nutzungsphase mit einbeziehen:

„Usability engineering is not a one-shot affair where the user interface is fixed up before the release of a product. Rather, usability engineering is a set of activities that ideally take place throughout the lifecycle of the product, with significant activities happening at the early stages before the user interface has even been designed.“

(Nielsen 1993, S. 71)

Inzwischen gibt es zahlreiche Varianten solcher (mehr oder weniger vollständigen) Lifecycles, die sich bezüglich des angesprochenen Problembereichs, bezüglich ihrer Integrationsbemühung im Hinblick auf Software-Entwicklungsprozesse und bezüglich ihrer Methoden- bzw. Prozessorientierung signifikant unterscheiden. Im folgenden sollen exemplarisch fünf der bekanntesten Usability Engineering Vorgehensmodelle kurz charakterisiert werden (Usability Engineering Lifecycle, Delta Method, Contextual Design, Scenario-based Development und Usage-Centered Design). Im Anschluss an jede Beschreibung wird eine knappe Beurteilung des jeweiligen Prozessmodells abgegeben.

3.1 Charakterisierung der bekanntesten Usability Engineering Vorgehensweisen

Die folgenden Vorgehensmodelle geben einen guten Überblick über den Umfang der adressierten Problemstellungen und die unterschiedlichen Herangehensweisen der Autoren. Im Anhang findet sich ein detaillierter Vergleich zwischen allen hier vorgestellten Prozessen (Anhang 11.1 Vergleich der verschiedenen Usability Engineering Prozessmodelle).

3.1.1 Der Usability Engineering Lifecycle (Mayhew 1999)

Der Usability Engineering Lifecycle ist ein vollständiges Usability Engineering Prozessmodell, dessen 16 Aktivitäten in die Phasen (1) Anforderungsanalyse, (2) Design / Testing / Development und (3) Installation unterteilt sind (vgl. Abbildung 5). Der Lifecycle ist angelehnt an das Software-Entwicklungsmodell OOSE (Object-Oriented Software Engineering) und explizit für eine Integration in einen Software-Entwicklungsprozess vorgesehen. Die Ergebnisse der Aktivitäten sollen in Form eines Style Guides dokumentiert werden.

Die Anforderungsanalyse spielt beim Usability Engineering eine zentrale Rolle und ist entscheidend dafür, ob die Software die Bedürfnisse der Benutzerinnen erfüllt oder nicht. Die Vorgehensweise für das Design ist ein strukturierter Top-down-Ansatz, der zunächst Anforderungen auf einer hohen Designebene festlegt und darauf basierend detaillierte Standards anwendet, um das Design zu komplettieren. Die Entwicklung findet in einem iterativen Prozess bestehend aus Design, Testing und Entwicklung statt, wobei zunächst mit einem Ausschnitt an Funktionalität gearbeitet wird, der nach und nach erweitert wird. Durch die Verwendung alternativer Techniken wird der Prozess flexibel, wobei das grundsätzliche Prozessmodell jedoch immer erhalten bleibt. Die optimale Umsetzung des Usability Engineering Lifecycles bedingt die volle Teilnahme eines multifunktionalen Teams. Der Prozess ist beendet, wenn nach der Installation und ersten User Feedbacks alle Mängel einer Software beseitigt sind und das Produkt voll funktionstüchtig und erfolgreich ausgebreitet ist.

1. **User Profiles** (Erhebung von Benutzerprofilen):

Benutzerprofile sind Zusammenstellungen spezifischer Benutzerinneneigenschaften, die relevant sind für das Design der Benutzerschnittstelle. Sie werden für die angestrebten Benutzer des zu entwickelnden Systems erhoben und sind massgebend für spätere Designentscheidungen. Ausserdem werden in diesem Schritt die wichtigsten Kategorien an Benutzerinnen identifiziert, die später in der Aufgabenanalyse untersucht werden sollen.

2. **Contextual Task Analysis** (Aufgabenanalyse im Kontext):

Ziel der Aufgabenanalyse ist es, ein benutzerzentriertes Modell der Arbeit zu erstellen, wie sie momentan von den angestrebten Benutzern ausgeführt wird. Dies geschieht am besten durch Beobachtungen der Benutzerinnen in ihrem Arbeitskontext. Aus den erstellten Arbeitsmodellen werden die funktionalen und qualitativen Anforderungen an das zu erstellende Produkt abgeleitet.

- 3. Usability Goal Setting** (Setzen von Usability Zielen):
Aus den Erkenntnissen der vorhergegangenen Aufgaben sowie aus den allgemeinen Geschäftszielen werden spezifische qualitative und quantitative Usability Ziele extrahiert, die massgebend sind für den Entwurf. Die Usability Ziele können zudem als Akzeptanzkriterien bei der Evaluation des Systems in späteren Usability Tests herangezogen werden.
- 4. Platform Capabilities and Constraints** (Erhebung von Plattformeigenschaften):
Bei diesem Schritt geht es darum zu klären, welche technischen Restriktionen bei der Systementwicklung gegeben sind, die den Freiraum bezüglich des Designs und der Interaktionstechniken einschränken.
- 5. General Design Principles** (Identifikation der grundlegenden Designprinzipien):
Für verschiedene Plattformen und Produkte kommen z.T. unterschiedliche Richtlinien für die Benutzerschnittstelle zum Tragen. Die relevanten und grundlegenden Designprinzipien müssen zunächst identifiziert und ausgewertet werden.
- 6. Work Reengineering** (Optimierung von Arbeitsabläufen):
Durch eine ausgewogene Berücksichtigung von bisherigen (gewohnten) Arbeitsabläufen und dem vorhandenen Automatisierungspotential werden die Arbeitsabläufe dahingehend optimiert, die Businessziele möglichst effektiv zu unterstützen, die Prozesse effizient zu gestalten und gleichzeitig den Lernaufwand für den Umgang mit dem neuen System so gering wie möglich zu halten.
- 7. Conceptual Model Design** (Erstellung des Systemkonzepts):
Die Erstellung eines kohärenten, regelbasierten Grobkonzeptes für die Benutzerschnittstelle stellt die erste Designaktivität im Usability Engineering Lifecycle dar. Das Systemkonzept soll zum einen die Erwartungen der zukünftigen Benutzerinnen erfüllen und zum anderen die Erstellung eines konsistenten Systems ermöglichen. In dieser Phase werden auf noch hohem Abstraktionsniveau die grundlegenden Präsentations- und Interaktionsregeln festgelegt und die wichtigsten Bildschirmmasken sowie die grundlegenden Navigationspfade identifiziert.
- 8. Conceptual Model Mock-ups** (Erstellung von einfachen Prototypen des Konzepts):
Um das Kernkonzept zu evaluieren und ein iteratives Vorgehen zu ermöglichen, werden einfache Prototypen gebaut, die die zentrale Funktionalität des zu entwickelnden Systems repräsentieren. Hierbei werden bereits die grundlegenden Präsentations- und Interaktionsregeln berücksichtigt.
- 9. Iterative Conceptual Model Evaluation** (Iterative Verbesserung des Konzepts):
Durch ein iteratives Vorgehen wird unter Einbeziehung von Benutzern das Konzept getestet und so lange verfeinert, bis es stabil ist und den Usability Anforderungen genügt.
- 10. Screen Design Standards (SDS)** (Festlegen von Bildschirmdesignstandards):
Um die spezifischen Anforderungen des zu entwickelnden Systems bedienen zu können, müssen vorhandene Styleguides (z.B. der verschiedenen Plattformen oder Firmenstyleguides) angepasst und/oder erweitert werden. Zusammen mit dem Systemkonzept bilden die SDS die Grundlage für das detaillierte und konsistente Design der Benutzerschnittstelle. Die SDS stellen die zweite Designaktivität des Usability Engineering Lifecycles dar.
- 11. Screen Design Standards Prototyping** (Erstellen von Prototypen der SDS):
Um die erweiterten Styleguides auf ihre Usability zu testen, werden sie in Form eines Prototypen visualisiert.
- 12. Iterative Screen Design Standards Evaluation** (Iterative Verbesserung der SDS):
Durch ein iteratives Vorgehen werden die SDS evaluiert und optimiert.
- 13. Style Guide Development** (Entwicklung des Produkt-Styleguides):
Der Produkt-Styleguide ist ein Dokument, in dem das Systemkonzept, die SDS und die Ergebnisse aller Aufgaben der Anforderungsanalyse zusammengefasst werden. Das Dokument soll in erster Linie dazu dienen, die bisherigen Erkenntnisse aus dem Usability Engineering Prozess innerhalb (und gegebenenfalls auch ausserhalb) des Projektteams zu kommunizieren.

14. Detailed User Interface Design (Detailliertes Design der Benutzerschnittstelle):

Auf der Basis der erarbeiteten Standards und der Designkonzepte werden detailliert alle Bildschirmmasken des Systems entworfen. Der detaillierte Entwurf der Benutzerschnittstelle ist die dritte Stufe der Designaktivitäten des Usability Engineering Lifecycles.

15. Iterative Detailed User Interface Design and Evaluation (Iterative Verbesserung des Designs der Benutzerschnittstelle):

Auch das detaillierte Design der Benutzerschnittstelle wird iterativ verfeinert, indem die wichtigsten Bereiche getestet und anhand der Testergebnisse optimiert werden.

16. User Feedback (Einholen von Benutzerfeedback):

Der Usability Engineering Lifecycle endet erst, nachdem usabilityrelevante Daten über konkrete Nutzungserfahrungen (z.B. in einer Pilotphase oder auch nach Ausbreitung eines Systems) vorliegen und die Usability des Produktes bestätigt wurde. Gibt es in der realen Nutzung noch Probleme mit der Usability des Produktes, müssen Verbesserungen vorgenommen werden.

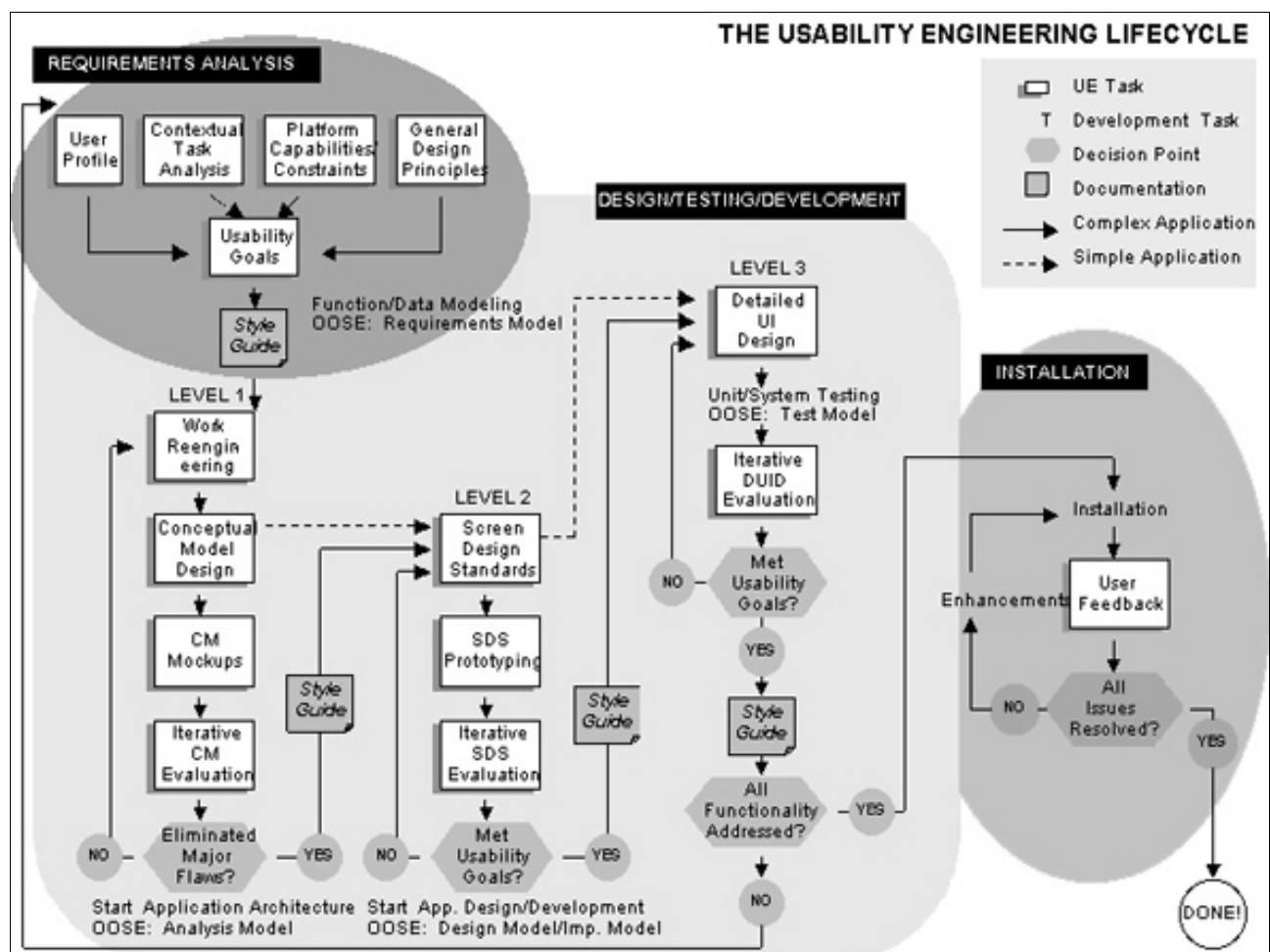


Abbildung 5: Der Usability Engineering Lifecycle

Der Usability Engineering Lifecycle ist in hohem Masse auf Kompatibilität mit einem modernen Entwicklungsprozess ausgelegt und kann daher theoretisch problemlos in einen objektorientierten Prozess eingebunden werden. Zu jeder der 15 Aktivitäten wird eine Vorgehensweise (Methode) vorgeschlagen, wobei häufig Alternativen in Form von Literaturverweisen und Abkürzungsmöglichkeiten genannt werden. Mayhew weist explizit darauf hin, dass der Prozess für jede Entwicklungsorganisation (je nach Grösse und vorhandenem Know-how) und jedes Entwicklungsprojekt (je nach Komplexität, Dauer und Projektteam) angepasst werden muss.

Die Besonderheiten des Usability Engineering Lifecycles im Vergleich zu anderen Prozessmodellen beschreibt Mayhew im Vorwort zu ihrem Buch folgendermassen:

„(..) the Usability Engineering Lifecycle (..) differs from those described by other authors in several important ways:

- It is more comprehensive and specific in its inclusion of Usability Engineering tasks in the overall lifecycle.
- It is more detailed in specifying how to **integrate usability tasks** into an already existing software development lifecycle.
- It is more detailed in its specification of how to carry out tasks and techniques in the lifecycle.
- In particular, it provides **more structure and detail** for the actual process of user interface design.“

(Mayhew 1999, S. X. Hervorhebungen: S.H.)

Vor allem der zweite und vierte Punkt erscheinen hinsichtlich der Erfolgchance für einen derartigen Prozess von Bedeutung.

3.1.2 Delta Method (WM-data AB & Ericsson Radio Systems AB 2000)

Delta ist ein Prozessmodell², das es ermöglichen soll, Computersysteme auf der Basis von Benutzerbedürfnissen zu entwickeln. Delta wurde von Ericsson (bzw. Infocom Consultants AB) in Zusammenarbeit mit der Universität von Linköping (Schweden) entwickelt. Mit Delta wurde der Überzeugung Rechnung getragen, dass es bei der Entwicklung von Computersystemen nicht nur um die Bereitstellung von Funktionalität geht, sondern dass die Systeme auch gebrauchstauglich (usable) sein müssen. Zur Verbreitung der Methode im Entwicklungsprozess von Ericsson und auch ausserhalb von Ericsson wurden Trainings und Kurse angeboten.

Der Delta Prozess lässt sich in 9 Phasen einteilen, wobei die Durchführung der Phasen z.T. iterativ ist (vgl. Abbildung 6). Für jede Phase wird angegeben, welcher Input zur Durchführung benötigt wird (und gegebenenfalls in der jeweiligen Phase überarbeitet wird → Iteration!). Für jede Phase werden Methoden vorgegeben, mit deren Hilfe Ergebnisse erarbeitet werden sollen, die wiederum als Input für die folgenden Phasen verwendet werden. Phase 5 – 8 sind explizit iterativ, wobei durchaus auch in den anderen Phasen Iterationen möglich sind (z.B. werden bei der Aufgabenanalyse ggf. die Benutzerprofile modifiziert).

1. System Definition (Systemdefinition):

In der ersten Phase versucht das Entwicklungsteam, eine grobe Vision von dem zu entwickelnden System zu erstellen, die sich an den Vorstellungen der zukünftigen Benutzerinnen orientiert. In der Regel haben die zukünftigen Benutzer bestimmte Probleme, die mittels des zu entwickelnden Systems gelöst werden sollen. Das System kann jedoch auch dazu dienen, bestimmte Arbeitsabläufe zu rationalisieren. In dieser ersten Phase werden ausserdem die verschiedenen Benutzerkategorien identifiziert, die als Grundlage für die anschliessend durchzuführende Benutzerprofil- und Taskanalyse dienen.

2. User Profiling (Erhebung von Benutzerprofilen):

In der Phase des User Profiling werden die bereits definierten Benutzerkategorien analysiert und z.B. bezüglich ihrer Computerkenntnisse, Professionalitätsgrad, Vorlieben und Fähigkeiten charakterisiert. Ausserdem werden die Probleme der Benutzerinnen mit den momentan vorhandenen Systemen untersucht.

3. Task Analysis (Aufgabenanalyse):

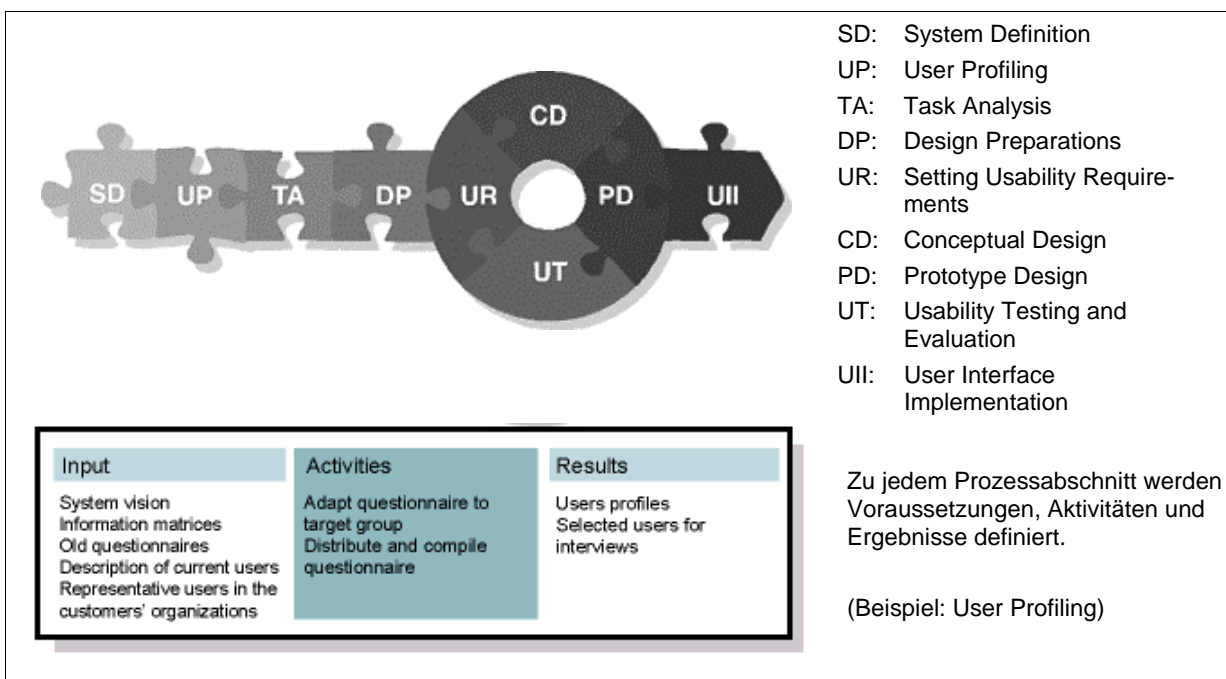
Das Ziel der Aufgabenanalyse ist es, aus erster Hand Kenntnisse über die Aufgaben der zukünftigen Benutzer zu erhalten. Hierzu werden einige repräsentative Benutzerinnen aus den verschiedenen Benutzerkategorien interviewt. Dadurch erhält man umfangreiche qualitative Daten über die konkreten Arbeitssituationen der verschiedenen Benutzergruppen. Die Ergebnisse der Aufgabenanalyse werden in Form von Aktivitätsdiagrammen dokumentiert, die später in Use Cases überführt werden können. Diese stellen wiederum die Grundlage für das Design des Systems und spätere Usability Tests dar.

4. Design Preparations (Designvorbereitungen):

In dieser Phase sammelt und analysiert das Entwicklungsteam alle erhobenen Daten der vorherigen Phasen und erstellt eine Liste mit Designratschlägen. Es handelt sich hierbei um das konsolidierte Wissen über die Benutzerinnen, deren Aufgaben und Arbeitskontext. Zusammen mit allfälligen Styleguides und anderem relevanten Material stellt dieses Ergebnis die Grundlage für das Interaktionsdesign dar.

² Im Verständnis dieser Arbeit handelt es sich bei Delta nicht um eine Einzelmethode, sondern um ein Prozessmodell. Vergleiche hierzu auch die Anmerkungen zur verwendeten Terminologie in dieser Arbeit auf Seite iv.

5. **Usability Requirements** (Usability Anforderungen):
 Die Phase der Usability Spezifikation mündet in einem Dokument, das die von allen Projekt-Stakeholdern akzeptierten Anforderungen an die zukünftige Usability des Systems definiert. Diese Anforderungen sind nach Möglichkeit zu quantifizieren, z.B. in Form von maximal benötigten Zeiten zur Durchführung bestimmter Aufgaben oder aber in Form von subjektiven Zufriedenheitswerten.
6. **Conceptual Design** (Erstellung des Designkonzepts):
 In dieser Phase finden die ersten konkreten Designaktivitäten statt. Die Erstellung des Designkonzepts kann als Zwischenschritt zwischen Analyse und Prototyping betrachtet werden. Das Ziel ist es, auf einem hohen Abstraktionsniveau ein Aufgabenmodell zu erstellen. Dieses soll später dazu dienen, die Arbeitsabläufe gut zu strukturieren, bevor man sich in Details auf der Ebene von Buttons und Menüs befindet. Hierzu werden Aufgaben gruppiert und zusammen mit den ihnen zugehörigen Objekten wird eine aufgabenorientierte Landkarte der gesamten Benutzerschnittstelle erstellt.
7. **Prototype Design** (Entwicklung eines Prototypen):
 Das Designkonzept und alle bisherigen Erfahrungen und Informationen aus den früheren Phasen dienen als Basis für Papierprototypen, mit denen erste Iterationen durchgeführt werden. Diese Form des Prototypings eignet sich hierfür besonders gut, da alle Projekt-Stakeholder (also auch die Benutzer) hierzu beitragen können. Zudem erlaubt Papier-Prototyping eine schnelle und effektive Visualisierung verschiedener Designoptionen. Nach der zweiten Iteration wird der Prototyp in der Regel computerbasiert weiterentwickelt.
8. **Usability Testing** (Test der Usability):
 Der Prototyp wird getestet. Es wird geprüft, ob er die geforderten Usability Ziele erfüllt. Während einige Benutzerinnen noch dabei sind, die Use Cases zu vervollständigen, wird bereits die Interaktion untersucht. Mit Hilfe eines Fragebogens kann nach dem Test die subjektive Einschätzung der Testbenutzer bezüglich des Prototyps abgefragt werden. Die Usability Tests werde zusammen mit den ermittelten Problemen und den Aussagen der Testbenutzerinnen dokumentiert.
9. **User Interface Implementation** (Implementation der Benutzerschnittstelle):
 Die Hauptaufgabe bei der Implementation ist es, den GUI Prototypen in eine voll funktionale Applikation zu transferieren. Es ist wichtig sicherzustellen, dass die Ergebnisse aus den Prototyping-Sessions und Usability Tests, aus der Anforderungsspezifikation und aus der Designspezifikation in das tatsächliche System überführt werden.



(WM-data AB & Ericsson Radio Systems AB 2000)

Abbildung 6: Prozess der Delta Method

Delta ist ein Prozessmodell, das in hohem Masse an die organisationellen Gegebenheiten und an den vorhandenen Software-Entwicklungsprozess bei Ericsson angepasst wurde. Es enthält nur Aktivitäten, Dokumente und Artefakte, deren Nutzen in ausreichendem Masse nachgewiesen ist. Die Benutzereinbeziehung wurde optimiert, da Benutzerverfügbarkeit als wertvolle Ressource erachtet wird. Die Vorgehensweise ist zielorientiert und baut auf einem guten pädagogischen Modell auf, das es erlaubt, Delta in kurzer Zeit zu erklären (z.B. einem Manager, der über die Ressourcenzuteilung zu entscheiden hat), wobei die Beschreibung der Vorgehensweise völlig undogmatisch ist. Delta wurde bei Ericsson in Form eines Paketes angeboten. Es gab etliche Services, einen Kurs, Broschüren und ein Handbuch darüber. Dies hat in hohem Masse dazu beigetragen, Delta verhältnismässig weit in der Entwicklergemeinschaft bei Ericsson zu verbreiten. Delta ist ein Beispiel für ein erfolgreiches Usability Engineering Prozessmodell. (Carlshamre und Rantzer 2001)

3.1.3 Contextual Design (Beyer und Holtzblatt 1998)

Das ausgesprochene Ziel von Contextual Design ist es, designrelevante Daten zu sammeln und diese auch tatsächlich als Designbasis zugänglich zu machen und zu verwenden. Das Herzstück von Contextual Design ist die Kontextanalyse (Contextual Inquiry), deren zugrundeliegende Philosophie besagt, dass die Struktur von Arbeit am besten am tatsächlichen Arbeitsplatz, also im Kontext, verstanden werden kann. Contextual Inquiry ist eine ethnographische Methode, bei der zukünftige Benutzerinnen des zu entwickelnden Systems bei ihrer täglichen Arbeit beobachtet und interviewt werden. Die gesammelten Informationen werden in Form von Modellen konsolidiert, die dazu beitragen sollen, sowohl den Kontext als auch die Arbeit selbst zu erklären. Insgesamt werden 5 Modelle erstellt, von denen 4 in hohem Masse für die Erfassung des Kontexts gebraucht werden: Ein Flussmodell (flow model) repräsentiert die einzelnen Aktivitäten und deren Koordination durch Kommunikation, in einem Sequenzmodell (sequence model) werden einzelne Aufgabenszenarien festgehalten, ein Artefaktmodell (artifact model) beschreibt die physikalischen Objekte näher, die in die Arbeitsabläufe involviert sind, ein Kulturmodell (cultural model) behandelt die Kultur in der Organisation und ein physisches Modell (physical model) konzentriert sich auf die physische Arbeitsumgebung.

Im folgenden sind die 8 Schritte des Contextual Design Prozesses kurz beschrieben (vgl. hierzu auch Abbildung 7):

- 1. Contextual Inquiry** (Untersuchungen im Kontext):
Der erste Schritt im Contextual Design dient dazu, die zukünftigen Benutzerinnen eines Systems zu verstehen: was sind ihre Wünsche, Bedürfnisse und Arbeitsansätze. Um genau herauszufinden, wer die Benutzer überhaupt wirklich sind und wie sie genau arbeiten, werden Interviews mit ihnen an ihrem Arbeitsplatz durchgeführt. So kann erkannt werden, wie die Benutzerinnen bei ihrer Arbeit vorgehen, warum sie welchen Schritt machen und welche Strategien sie bei der Erledigung der Arbeit verfolgen. Beim Interview wird das Beziehungsmodell Meister – Lehrling angewendet. Durch Diskussion der Beobachtungen gelangen Benutzer und Interviewer zu einem gemeinsamen Verständnis der Arbeit.
- 2. Team Interpretation Session** (Interpretation im Team):
In einem Team aus Mitgliedern verschiedenster Funktionen im Entwicklungsprozess wird das Interview gemeinsam interpretiert, um Einsichten in relevante Aspekte der Arbeit und mögliche Problempunkte zu erhalten. Jedes Teammitglied kann hierbei seine eigene Perspektive (z.B. Marketing, Design, Geschäftsprozesse usw.) beisteuern.
- 3. Work Modeling** (Modellierung der Arbeit):
Arbeitsabläufe sind komplex und voller Details, die man meistens nicht leicht erfassen kann. Zudem ist es für Aussenstehende schwierig, die Struktur der Arbeit von anderen zu erkennen und hinter die Kulissen zu blicken, um Intentionen, Strategien und Motivationen zu durchschauen. In Arbeitsmodellen können sowohl die Arbeit des einzelnen als auch ganze Arbeitsprozesse in Organisationen abgebildet werden. Fünf verschiedene Modelle liefern fünf Perspektiven auf die Arbeit.
- 4. Consolidation** (Konsolidierung der gesammelten Informationen):
Systeme werden nicht für Einzelpersonen entwickelt, sondern für die Gesamtheit der Benutzerinnen. Mit Hilfe einer induktiven Vorgehensweise (vom Konkreten zum Allgemeinen) werden die Gemeinsamkeiten der Arbeit verschiedener Leute ermittelt. Der Konsolidierungsprozess vereint die Daten aus den individuellen Interviews zu einem gemeinsamen Modell, so dass das Team allgemeine Muster und Strukturen erkennen kann, ohne jedoch die individuellen Variationen zu verlieren. Das Affinitäts-Diagramm bringt all diese Aspekte und Einsichten über alle Benutzer in ein einziges, hierarchisches Diagramm zusammen.

Die konsolidierten Arbeitsmodelle zeigen ebenfalls die gemeinsamen Strategien der Benutzerinnen bei der Erledigung ihrer Arbeit auf. Zusammen mit dem Affinitäts-Diagramm ergeben die konsolidierten Arbeitsmodelle ein vollständiges Bild auf die Zielgruppe, für die die Software entwickelt werden soll. Sie geben so dem Team einen Fokus für das Design sowie für die Kommunikation darüber und zeigen, wie die Arbeit zusammenhängt.

5. **Work redesign** (Optimierung der Arbeitsabläufe):

Das Designteam entwirft eine neue Systemstruktur, die die Arbeitprozesse der Customer verbessert. Die bisher erarbeiteten, konsolidierten Daten sind die Grundlage für Überlegungen, wie mit Hilfe technologischer Möglichkeiten die neuen Arbeitsprozesse unterstützt werden können. Die neu strukturierten Arbeitsprozesse werden in Beschreibungen konkreter Arbeitsabläufe von Customern zusammengefasst (Visions), die das System sowie Supportstrukturen und alles beinhalten, was zum Erfolg der neuen Arbeitspraxis beiträgt.

6. **User environment design (UED)** (Entwurf eines Systemkonzeptes aus Benutzersicht):

Das neue System muss eine geeignete Funktionalität aufweisen, um einen reibungslosen Arbeitsablauf zu garantieren. Hierzu müssen die Designer einen genauen Plan ihres Systems ausarbeiten, der die Antwort auf die Customer Bedürfnisse darstellt und sich hinter den verschiedenen Interfaces verbirgt. Im User Environment Design wird auf der Grundlage der Storyboards dieser Plan des Systems entwickelt. Es wird aufgezeigt, wie jeder einzelne Teil des Systems eine bestimmte Arbeit im Arbeitsprozess des Customers unterstützt, welche Funktionalität vorhanden ist und wie der Customer zwischen den verschiedenen Systemkomponenten navigiert, ohne jedoch konkrete Interfaces zu betrachten. Das UED dient als Basis für alle Entwicklungsarbeiten und garantiert die Kohärenz des gesamten Systems. Mit seiner Hilfe kann sichergestellt werden, dass das System den Anforderungen der User entspricht, und es unterstützt zudem die Projektplanung und –strategie.

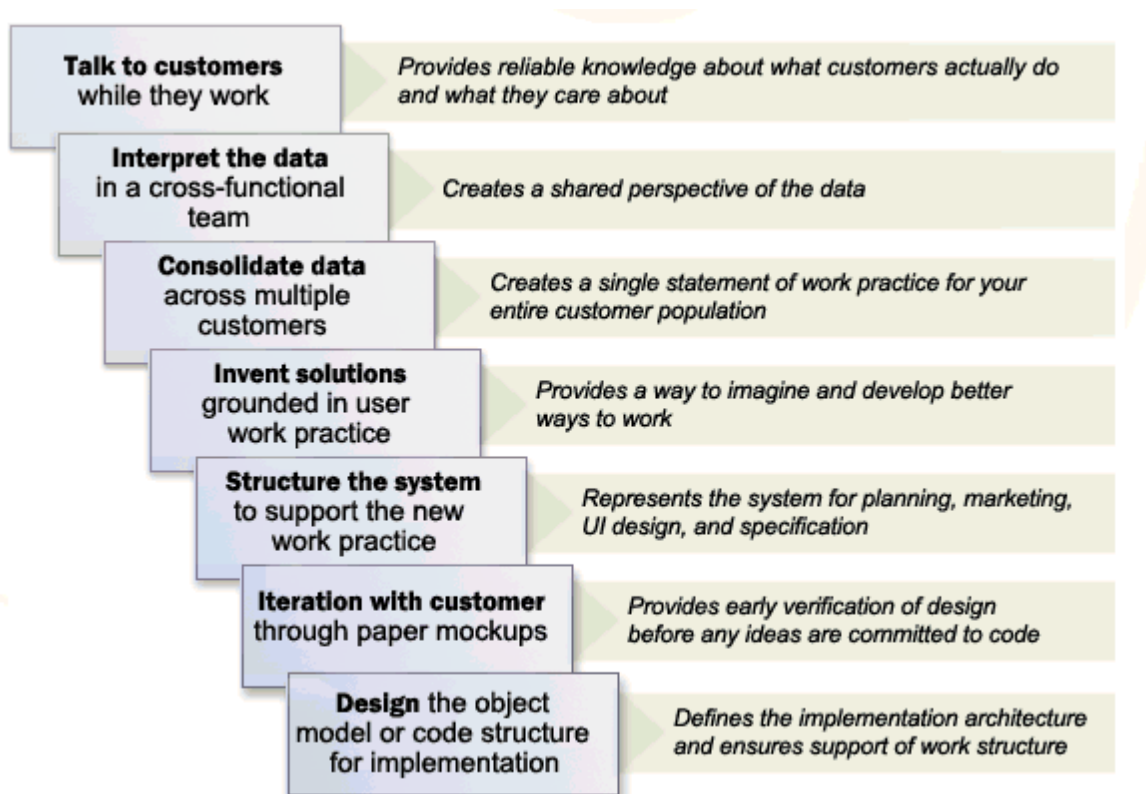
7. **Mock-up and test with customers** (Prototyping und Test):

Je früher Probleme eines Systems erkannt werden, desto billiger wird deren Behebung. Daher ist das frühe Testing – bevor die Ideen in Code umgesetzt werden – ein wichtiger Schritt bei der Entwicklung eines Systems. Je einfacher die Mittel sind, mit denen getestet wird, desto mehr Iterationsschritte sind möglich. Deshalb ist es sinnvoll, Papierprototypen für die ersten Tests zu verwenden. Das Prototyping wird zum eigenständigen Design Hilfsmittel, mit dem Designideen kommuniziert werden und Customer direkt in den Designprozess eingebunden werden können. Durch dieses Vorgehen kann die Struktur bei jedem Entwicklungsschritt so lange bearbeitet werden, bis das System tatsächlich den Customeranforderungen genügt. Die so entwickelte Struktur dient schliesslich als Basis für das detaillierte User Interface Design.

8. **Priorization and object-oriented design** (Prioritätensetzung und Objektorientiertes Design):³

Design ist der erste und wichtigste Schritt bei der Entwicklung eines neuen Systems. Ein stabiles, konsistentes Systemdesign ist für das Projektteam eine brauchbare Grundlage, auf deren Basis die Implementation des Systems geplant werden kann. Prioritätensetzung hilft bei der Planung verschiedener Releases. Dabei wird das System in einzelne Einheiten aufgeteilt, die nach und nach implementiert werden können. Jede dieser Einheiten für sich hat ihren definierten Wert für die Customer. Der Schritt vom System Design zur Implementation erfolgt am leichtesten durch objektorientiertes Design. Ausgehend vom User Environment Design und den Storyboards werden einzelne Anwendungsbeispiele in ein Objekt design überführt, mit dessen Hilfe der traditionelle Implementationsprozess durchgeführt werden kann.

³ Schritt 8 des Prozesses ist nicht sehr ausführlich im Buch Contextual Design enthalten. Beyer und Holtzblatt erwähnen jedoch diesen Schritt explizit in der Beschreibung des Prozessablaufs auf der Homepage ihrer Firma *InContext Enterprises, Inc.* unter <http://www.incent.com/>.



(Diagramm von InContext Enterprises, <http://www.incent.com/cd/cdhow.html>)

Abbildung 7: Prozess des Contextual Designs

Die Stärke von Contextual Design liegt in seinen detaillierten Anleitungen, die neben den eigentlichen Methoden zur Durchführung der Kontext- und Aufgabenanalysen und der Erstellung eines Designkonzeptes auch zahlreiche Techniken beinhalten, die die zwischenmenschliche Dimension von Design in multifunktionalen Teams berücksichtigen. Bezüglich des gesamten Prozesses zur Entwicklung eines Softwareproduktes ist die Reichweite von Contextual Design auf die Phase der Anforderungsspezifikation und der Konzeptentwicklung begrenzt. Problematisch am Buch Contextual Design ist die eher schwache Strukturierung. Die Beschreibung von Techniken, Anwendungsbeispiele und Aspekte der Umsetzung in realen Projekten werden bunt gemischt. Die Darstellung ist wenig geeignet, sich ein mentales Modell von der Vorgehensweise zu bilden. Dies erschwert es, die zweifellos wertvollen Hinweise und interessanten Methoden wirklich zu verstehen. Das grundlegende Problem bei Contextual Design ist jedoch, dass es sehr vieler organisationaler Voraussetzungen bedarf, um die vorgeschlagenen Methoden tatsächlich anwenden zu können. Nahegelegene Anwender sind eventuell Organisationen, in denen marktorientiert Software entwickelt wird und die von der Innovationsförderung des Contextual Design Prozesses profitieren. Nicht ganz zufällig ist daher SAP eine der grossen Kundinnen von Bayer/Holtzblatt. Die Widerstände gegen Contextual Design sind jedoch angeblich beachtlich⁴ und wer in jüngerer Zeit ein Produkt von SAP zu Gesicht bekommen hat, darf daran zweifeln, dass bei SAP weitreichende benutzerzentrierte Entwicklungsprozesse zur Anwendung kommen. Dies ist vielleicht nicht zuletzt auf die hier erwähnten Probleme des Prozesses zurückzuführen. Alle Themen zur Unterstützung der konkreten Bildschirmgestaltung in Bezug auf Layout, Verwendung bestimmter Interaktionselemente usw. bleiben im Contextual Design unbehandelt. Diese tragen jedoch ebenfalls signifikant zur Usability eines Systems bei, weshalb neben Contextual Design auf jeden Fall noch weitere Usability Engineering Aktivitäten zum Einsatz kommen sollten.

⁴ Aussage eines SAP Mitarbeiters aus einem Entwicklungsteam in einem Gespräch auf der „Mensch & Computer 2001“

3.1.4 Scenario-based Development (Rosson und Carroll 2002)

Scenario-based Development ist ein Ansatz, der ein grundlegendes methodisches Vorgehen zur Ausübung von Usability Engineering Aufgaben beschreibt. Wie der Name schon sagt, ist die Kernidee des von Rosson und Carroll vorgestellten Rahmenwerks ein Usability Engineering Prozess, bei dem Szenarien die zentrale Repräsentation von Analyse und Design darstellen (als Ergänzung oder anstelle anderer Dokumentationen oder Modelle). Dabei sind die Szenarien nicht nur als Ergebnisse (Artefakte) zu verstehen, sondern auch als Methode und Hilfsmittel zur Verdeutlichung und Diskussion von Problem- und Lösungsansätzen. Szenarien beschreiben ein existierendes oder geplantes System aus der Perspektive eines oder mehrerer realer bzw. realistischer Benutzer. Sie umfassen die Ziele, Pläne und Reaktionen der Benutzerinnen und eignen sich hervorragend, um [Trade-offs](#) zu verdeutlichen, die im Rahmen jeder GUI-Entwicklung zu behandeln und zu entscheiden sind. Zudem haben sie den Vorteil, dass sie von allen Projektbeteiligten gleichermassen verstanden werden und so alle am Designprozess teilhaben und Lösungsvorschläge diskutieren können. Parallel zu den Szenarien werden sogenannte Claims entwickelt. Sie stellen das zweite wichtige Element von Scenario-based Development dar. Die Kernelemente, die in einem Szenario angesprochen werden, werden zusammen mit ihren (hypothetischen) Vor- und Nachteilen aufgelistet. So wird deutlich, welche Auswirkungen jedes Element auf die Usability des Systems hat. Überwiegen die Nachteile eines Elements dessen Vorteile, so wird es in der Regel verworfen.

Die Aktivitäten des Scenario-based Development lassen sich in die Phasen Anforderungsanalyse, Design, Prototyping und Evaluation unterteilen, wobei der Designschritt noch weiter in Aktivitätsdesign, Informationsdesign und Interaktionsdesign differenziert wird. Die Vorgehensweise ist iterativ. Im folgenden werden die Phasen zusammen mit den ihnen zugehörigen Methoden und Ergebnissen kurz beschrieben. Abbildung 8 zeigt einen Überblick über das szenarien-basierte Entwicklungsmodell.

1. Analyzing Requirements (Anforderungsanalyse):

Im Rahmen der Anforderungsanalyse wird mittels verschiedenster Methoden untersucht, welche Benutzer oder sonst wie betroffenen Personen bezüglich des zu entwickelnden Systems vorhanden sind, wie deren Aufgaben und Arbeitskontext aussieht und welche Bedürfnisse die zukünftigen Benutzerinnen bezüglich des Systems haben. Als Methoden kommen z.B. eine Stakeholderanalyse, Aufgabenanalysen, Benutzerprofile, Interviews, Feldstudien usw. in Frage. Die Ergebnisse dieser ersten Phase werden in Form von Problemszenarien festgehalten. Problemszenarien sind Geschichten über die momentane Arbeitspraxis. Diese werden sorgfältig entwickelt, um solche Aspekte aus dem Alltag der Stakeholder und deren Aktivitäten aufzuzeigen, die Einfluss auf das zu entwickelnde System haben. Zu den Szenarien werden sogenannte Claims erstellt. Claims dokumentieren die *Trade-offs* bezüglich wichtiger Merkmale der erstellten Szenarien. Achtung: Problemszenarien und Claims sind noch nicht die Anforderungsspezifikation. Sie enthalten jedoch implizit die Anforderungen an das System.

2. Activity Design (Aktivitätsdesign):

In dieser ersten Designphase, werden auf der Basis vorhandener Probleme und Möglichkeiten neue Aktivitäten entworfen, um die Probleme zu lösen. Als Methoden kommen z.B. [Claimanalysen](#), Partizipatives Design, Metaphernbildung usw. in Frage. Das Ziel dieser Phase ist es, die funktionalen Anforderungen an das System zu spezifizieren. Hierbei muss besonderer Wert darauf gelegt werden, leicht verständliche Aktivitäten zu entwerfen, die es den zukünftigen Benutzerinnen erlauben, ein funktionierendes mentales Modell des Systems zu entwickeln. Die Funktionalität des Systems muss sowohl die persönlichen Ziele der Benutzer befriedigen als auch die möglicherweise vorhandenen Ziele des sozialen Gefüges, in dem das System eingesetzt wird. Die Ergebnisse dieser ersten Designphase werden in Form von Aktivitätsszenarien und Claims festgehalten. Aktivitätsszenarien enthalten Ideen über bestimmte Systemfunktionalitäten, mit denen sich die Probleme aus den Problemszenarien lösen lassen. An dieser Stelle werden zunächst nur die Funktionalitäten im Hinblick auf die Ziele und Aktivitäten der Benutzerinnen beschrieben und noch keine Details über die Präsentation der Information oder die physische Interaktion ausformuliert. Scenario-based Development geht von ständigen Iterationen aus, in denen Designideen mit Hilfe der Szenarien und Claims ausprobiert, überarbeitet und verfeinert werden.

3. Information Design (Informationsdesign):

In dieser Phase wird die visuelle Repräsentation der Objekte und möglichen Aktionen des neuen Systems entworfen und in einer Art und Weise arrangiert, dass die Wahrnehmung und das Verstehen der angebotenen Information möglichst leicht gemacht wird. Informationsdesign umfasst z.B. den Entwurf des generellen Bildschirmlayouts, die Definition einzelner Pages in einem Webprojekt, das Design von Menüs und Dialogboxen sowie die Gestaltung von Icons. Ziel ist es, den Benutzern die Auswertung der

angebotenen Information möglichst leicht zu machen. Dies geschieht durch die Unterstützung der Wahrnehmung, Interpretation und Sinnbildung der Information (Überbrückung des sogenannten „Gulf of Evaluation“ nach Norman 1986). Dies wird erreicht durch korrekte Umsetzung von Gestaltungsprinzipien, durch das Angebot und die Sichtbarmachung von sinnvollen Interaktionsmöglichkeiten, durch die Verwendung von Metaphern und durch konsistentes Design. Spezielle Ergebnisse dieser Phase sind Informationsszenarien und Claims. Hierzu werden die Aktivitätsszenarien sukzessive um Details erweitert über das, was die Benutzerinnen sehen (bzw. hören oder fühlen, je nach Applikation). Ausserdem wird festgehalten, wie die Benutzer die angetroffene Information wahrnehmen, interpretieren und mit Sinn füllen. Spätestens zu diesem Zeitpunkt ist es sinnvoll, die ausformulierten Szenarien durch Skizzen zu ergänzen, indem z.B. einige der zentralen Screens mit all ihren visuellen Elementen entworfen werden.

4. **Interaction Design** (Interaktionsdesign):

Beim Interaktionsdesign geht es darum, die Mechanismen zu spezifizieren, über die die Benutzerin Zugang und Manipulationsmöglichkeiten zu der angebotenen Information erhält. Hierbei muss sichergestellt werden, dass der Benutzer zur richtigen Zeit das richtige mit dem System tun kann. Interaktionsdesign dient dazu, den sogenannten „Gulf of Execution“ zu überbrücken (Norman 1986). Das heisst, die Benutzerin muss ihr persönliches Ziel in die Logik des Systems übersetzen (Systemziel), sie muss erkennen können, welche Schritte notwendig sind, um dieses Systemziel zu erreichen und wissen, wie die einzelnen Schritte tatsächlich ausgeführt werden können. Wichtige Aspekte beim Interaktionsdesign sind beispielsweise die Abbildung der Realweltobjekte im System (in Form von Metaphern, Icons, Text), die Art und Weise, wie diese Objekte angeboten werden (immer sichtbar, über ein Menü zugänglich), wie mit ihnen interagiert werden kann (direkt manipulierend, über Kommandos usw.). Komplexe Abläufe müssen in handhabbare Einheiten zerlegt werden, um die Interaktion zu erleichtern und für die eigentliche Aktion müssen die geeigneten Eingabemedien (Maus, Tastatur, Joystick) zur Verfügung gestellt werden. Bei all diesen Aspekten ist es wichtig, dass das System ausreichendes Feedback gibt, um den Benutzer bei der Ausführung zu unterstützen. Die Ergebnisse des Interaktionsdesigns sind wiederum Szenarien, die nun die detaillierte Interaktion zwischen Benutzerin und System beinhalten, und Claims zu den verschiedenen [Features](#). Zusätzlich werden einzelne Interaktionsabläufe zur besseren Übersichtlichkeit in Form von Storyboards skizziert, wobei Storyboards im Prinzip bereits einen ersten groben Prototypen des Systems darstellen.

5. **Prototyping** (Entwicklung von Prototypen):

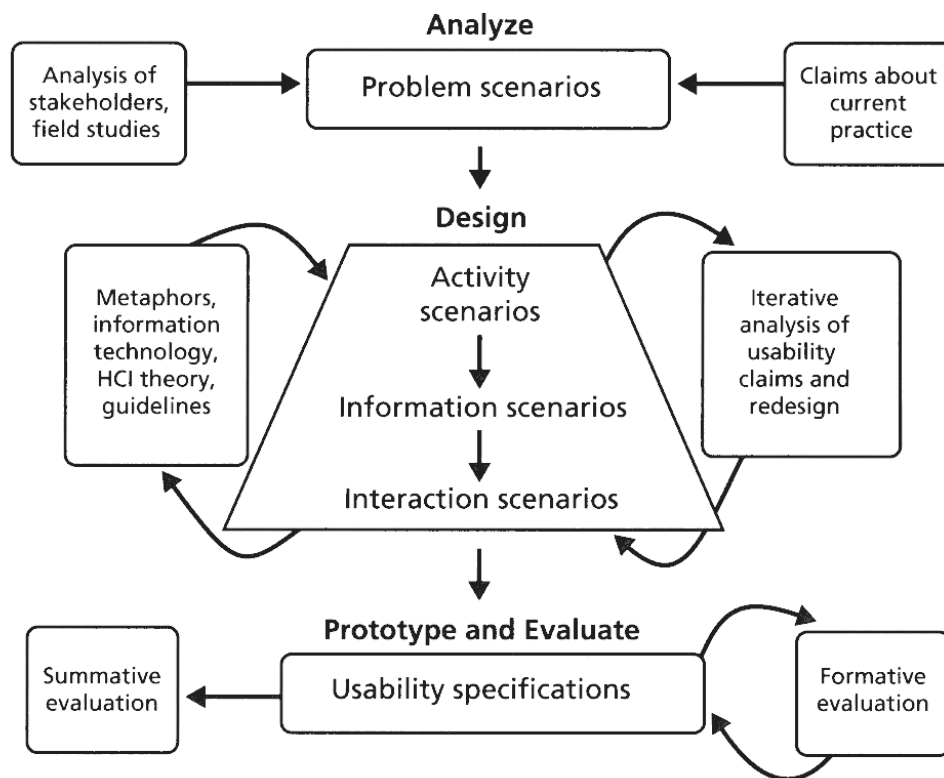
Ein wichtiges Element iterativer Entwicklungsprozesse ist die Entwicklung von Prototypen und deren Evaluation auf deren Basis ein Redesign und später letzte Verbesserungen stattfinden können. Prototypen sind konkrete, wenn auch nur teilweise Implementationen eines Systems und dienen zur Beantwortung der unterschiedlichsten Fragestellungen (nicht nur bezüglich des Benutzerinterfaces, sondern z.B. auch bezüglich der technischen Realisierbarkeit oder Performanz). Mit Hilfe von Prototypen können strittige Fragen im Zusammenhang mit Designvorschlägen beantwortet werden, in dem alternative Lösungen realisiert und miteinander verglichen werden. Szenariobasierte Argumente und *Claimanalysen* unterstützen den Prototypingprozess, indem sie repräsentative Benutzerinnen beschreiben und sinnvolle Testbedingungen und Testfälle definieren. Je nach Projektstand können einfache (z.B. auf Papier) oder voll funktionale (z.B. HTML oder Visual Basic) Prototypen entwickelt werden.

6. **Usability Evaluation** (Prüfung der Usability):

Hinter dem Begriff Usability Evaluation verbirgt sich eine Vielzahl verschiedener Methoden, die alle zum Ziel haben, Feedback für die Software-Entwicklung bereitzustellen, um einen iterativen Entwicklungsprozess zu unterstützen. Evaluationen dienen zum einen dazu, Input für eine Verbesserung des Designs einer Benutzerschnittstelle zu erhalten oder aber, um die Qualität eines fertigen Produktes zu messen. Hierzu stehen analytische oder empirische Methoden zur Verfügung, wobei in der Regel empirische Methoden im Usability Engineering vorgezogen werden, da sie reale Benutzer einbeziehen. Analytische Methoden haben jedoch den Vorteil, dass sie z.T. mit geringem Aufwand durchzuführen sind und daher eher akzeptiert werden (z.B. die von Nielsen 1994b vorgeschlagene heuristische Evaluation). In jedem Fall ist es anzustreben, die Usability Eigenschaften in bezug auf eine definierte Usability Spezifikation zu testen. Carroll und Rosson schlagen vor, die Usability Spezifikation als parallelen Prozess zu der funktionalen Spezifikation eines Systems durchzuführen. Hierdurch soll sichergestellt werden, dass Usabilityaspekte immer berücksichtigt werden, wenn neue *Features* in den Entwurf integriert werden sollen. Beim Scenario-based Development wird die Usability Spezifikation direkt aus den Designszenarien abgeleitet.

7. User Documentation (Erstellung eines Benutzungsdokumentes):

Es gibt zahlreiche verschiedene Formen von Hilfestellungen für die Benutzung eines Systems. Üblicherweise finden sich Handbücher, Onlinehilfen, Tutorsysteme und seltener auch Benutzerforen. Hilfesysteme können kontextsensitiv (z.B. auch in Form von Fehlermeldungen) oder allgemein sein (wie z.B. die beliebten FAQs). Benutzer ziehen in aller Regel ‚learning by doing‘ dem Lesen von Handbüchern vor. Rosson und Carroll schlagen ein Modell für ein Hilfesystem vor, das sie ‚Minimalism‘ nennen. Dieses Dokumentationsmodell basiert auf folgenden Kernprinzipien: 1.) Die Dokumentation soll aktionsorientiert sein, d.h. die Benutzerinnen sollen kontinuierlich dazu ermuntert werden, tatsächlich mit dem System zu arbeiten und dabei direkte Hilfestellungen erhalten. 2.) Die Dokumentation soll entsprechend der tatsächlichen Aufgabenstellungen strukturiert sein, die den Benutzern aus ihrem alltäglichen Kontext bekannt sind. 3.) Die Dokumentation soll Fehler antizipieren und behandeln, d.h. häufige Fehleingaben sollen vom System erkannt und automatisch korrigiert werden. Übliche konzeptionelle Fehler sollen so behandelt werden, dass den Benutzerinnen anhand der Fehlerbehandlung ein besseres Verständnis des Systems nähergebracht werden kann. Wie jede andere Designentscheidung hat auch jede Art von Benutzerhilfe ihre spezifischen *Trade-offs*, die bei der Auswahl des am besten geeigneten Hilfesystems berücksichtigt werden müssen.



(Rosson und Carroll 2002, S. 25)

Abbildung 8: Überblick über ein Prozessmodell auf der Basis von Szenarios

Scenario-based Development gibt ein Rahmenwerk für die Ausübung von Usability Engineering Aktivitäten vor, bei dem die Aushandlung von *Trade-offs* im Mittelpunkt steht. Diese Vorgehensweise hat den Vorteil, dass rationale Designentscheidungen auf der Basis hypothetischer oder bekannter Vor- und Nachteile einzelner Designelemente gefällt werden können. Scenario-based Development definiert nahezu ausschließlich den Aspekt der Software-Entwicklung, der sich auf die Benutzerschnittstelle bezieht und beschreibt einen eigenständigen Prozess ohne den Anspruch auf Integration in den vorhandenen Software-Entwicklungsprozess einer Organisation. In einem kurzen Abschnitt über die Methode POV (Point of View) wird eine Annäherung an die objektorientierte Software-Entwicklung gesucht, in dem zentrale Designobjekte identifiziert, analysiert und beschrieben werden. Probleme bezüglich fehlender Kompatibilität dieses Usability Engineering Prozesses mit dem vorhandenen Software-Entwicklungsprozess werden jedoch nicht adressiert.

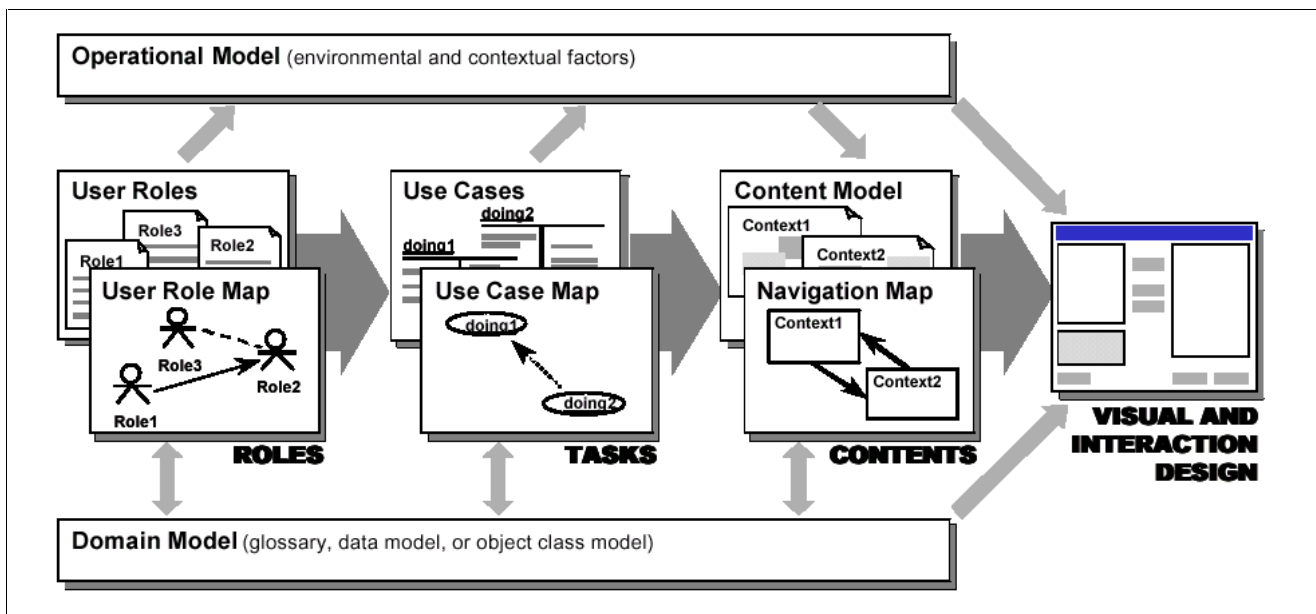
3.1.5 Usage-Centered Design (Constantine und Lockwood 1999)

Die bisher dargestellten Usability Engineering Prozessmodelle und Methodologien haben die Benutzerin ins Zentrum der Betrachtung gestellt. Usage-Centered Design setzt den Fokus stattdessen auf die **Benutzung** eines Softwareproduktes. Dieser Unterschied mag subtil erscheinen, hat jedoch nicht zuletzt zum Ziel, an die Tradition der Use Cases aus der objektorientierten Software-Entwicklung anzuknüpfen. Constantine und Lockwood halten die herkömmlichen Use Cases (nach Jacobson 1995) für die Modellierung von User Interface Komponenten und die Interaktion zwischen Benutzerinnen und diesen Komponenten nicht für optimal. Stattdessen wird im Usage-Centered Design der «essential use case» mit einer eigenständigen, sehr strukturierten Notation eingeführt. Essential Use Cases sind auf ein absolutes Minimum beschränkt, wodurch sie in hohem Masse unabhängig von einer konkreten Lösung sind. Sie stellen das Zentrum des Usage-Centered Designs dar:

„A key to enhancing usability is to defer user interface design details by expressing requirements independent of their implementation and of their realization in a particular form of user interface.“ (Constantine und Lockwood 2000, S. 7)

Diese systematische Trennung zwischen Problembeschreibung (z.B. funktionalen Anforderungen) und Problemlösung (z.B. Lösungsvorschläge in Form von Prototypen) ist eine Eigenheit des Usage-Centered Designs, die besonders hervorzuheben ist und zahlreiche Probleme im Entwicklungsalltag lösen hilft.

Usage-Centered Design ist ein modell-getriebener Prozess, bei dem primär 3 abstrakte Modelle zur Anwendung kommen: Das Benutzerrollen Modell (User Role), das Aufgabenmodell (Use Cases) und das Inhaltsmodell (Content Model). Jedes dieser 3 Modelle besteht aus 2 Teilen, einer Sammlung von Beschreibungen und einer Landkarte der Beziehungen zwischen diesen Beschreibungen (User Role Map, Use Case Map und Navigation Map), siehe Abbildung 9.



(Constantine und Lockwood 2000)

Abbildung 9: Zusammenhang zwischen Modellen des Usage-Centered Designs

Usage-Centered Design ist eine Sammlung von koordinierten Aktivitäten, die dazu beitragen sollen, die Usability von Softwaresystemen systematisch zu erhöhen. Die beschriebenen Aktivitäten umfassen zentrale Aufgaben der Software-Entwicklung auf der Ebene der Konzeption und des Designs und sind in hohem Masse tauglich, in einen Standard-Software-Entwicklungsprozess integriert zu werden. Im folgenden soll eine kurze Beschreibung des Usage-Centered Designs gegeben werden:⁵

⁵ Die in ihrem Buch „Software for Use“ (1999) vorgestellten Techniken, haben Constantine und Lockwood in den letzten Jahren im Rahmen verschiedener Arbeiten weiter erläutert und ausgefeilt. Auf ihrer Homepage <http://www.forUse.com> bieten sie zahlreiche Informationen und Dokumente zum kostenlosen Download an.

1. **User Role Modeling** (Modellierung von Benutzerrollen):
Der erste Schritt in der Phase der Anforderungsspezifikation bezieht sich auch im Usage-Centered Design auf die Benutzerinnen. Die Modellierung von Benutzerrollen hat zum Ziel, ein kommunizierbares Verständnis der Benutzer zu erlangen. Constantine und Lockwood erwähnen diese Aktivität zwar nicht in ihrem Aktivitätsdiagramm, widmen diesem Schritt jedoch ein eigenes Kapitel. Hier geht es vor allem darum, den Entwicklerinnen die Bedeutung der Ziele realer Benutzer zu vermitteln und ihnen nahezu-bringen, dass Entwurfsentscheidungen bei der Systementwicklung in erster Linie auf Kenntnissen über die Benutzerinnen des Systems basieren müssen. Ergebnisse der Modellierung von Benutzerrollen sind zum einen die strukturierten Rollenmodelle und zum anderen eine Übersichtskarte über alle relevanten Benutzer.
2. **Collaborative Requirements Dialog** (Anforderungsspezifikation):
Das Aktivitätsmodell des Usage-Centered Designs beginnt mit einem Trio aus Aktivitäten, die dazu dienen sollen, die grundlegenden Anforderungen an das System zu definieren. Die erste Aktivität in diesem Zusammenhang ist ein gezielter Dialog und Verhandlungen zwischen den Entwicklerinnen und den zukünftigen Benutzern des Systems, um die Anforderungen an das System zu erheben.
3. **Domain Modeling** (Modellierung der Anwendungsdomäne):
Die Modellierung der Anwendungsdomäne liegt laut Constantine und Lockwood streng genommen ausserhalb des Anwendungsbereiches von Usage-Centered Design, zeigt jedoch, wie eng die Vorgehensweise mit einem objektorientierten Entwicklungsprozess verknüpft ist. Die Modellierung der Anwendungsdomäne hat zum Ziel, eine Abbildung aller Konzepte und Konstrukte, die mit dem System in Zusammenhang stehen, zu entwickeln. Die Anwendungsdomäne wird heute oft in Form eines Klassenmodells modelliert. Bei diesem Schritt geht es vor allem darum, das Vokabular des Systems und seiner Funktionen zu definieren.
4. **Task Modeling** (Aufgabenmodellierung):
Das Zentrum der Anforderungsphase und gleichzeitig des gesamten Usage-Centered Designs stellt die Aufgabenmodellierung dar. Ziel dieser Aktivität ist es, zusammen mit dem Rollenmodell ein genaues und vollständiges Bild der Arbeit zu beschreiben, die durch das neue System unterstützt werden soll. Constantine und Lockwood verwenden zur Anforderungsmodellierung «essential use cases», die auf den Intentionen der Benutzer basieren, anstatt genau zu beschreiben, wie die konkreten Schritte oder Mechanismen ausgeführt werden, um diese Intentionen zu erfüllen. Durch die Vermeidung impliziter Annahmen und ihre Lösungsneutralität erlauben Essential Use Cases eher als die herkömmlichen Use Cases die Entwicklung wirklich innovativer Lösungswege. Ergebnisse der Aufgabenmodellierung sind die einzelnen Use Cases und ein Übersichtsplan über alle vorhandenen Use Cases.
5. **Interface Content Modeling** (Modellierung des Inhalts der Benutzerschnittstelle):
Bei der Modellierung des Inhalts der Benutzerschnittstelle werden die verschiedenen Arbeitskontexte definiert, aus denen sich das System zusammensetzt. Einzelne Funktionen werden diesen Kontexten zugeordnet und die Navigation zwischen den Arbeitsbereichen wird definiert. Beim Inhaltsmodell handelt es sich um ein abstraktes Modell, das zwar abstrakt, aber ganz präzise alle Werkzeuge und Materialien, aus denen sich das System zusammensetzt auf der Benutzerschnittstelle verteilt. Die Organisation des Inhalts der Benutzerschnittstelle orientiert sich an dem Übersichtsplan der vorhandenen Use Cases. Das Ergebnis dieses Arbeitsschrittes ist ein Kontext-Navigationsplan über das gesamte System.
6. **Implementation Modeling** (Implementationsmodellierung):
Mit Implementationsmodellierung ist der Prozess des detaillierten Designs und Prototypings des Systems gemeint. Hierbei wird einerseits das Layout definiert und andererseits die konkrete Interaktion der Benutzerinnen mit dem System festgelegt. Bei der Erstellung des Implementationsmodells muss die Designerin der Benutzerschnittstelle entscheiden, wie die abstrakten Interaktionsräume des Inhaltsmodells in einem konkreten GUI realisiert werden, z.B. welche Screens es gibt, welche Interaktionselemente verwendet werden, wie die Dialoge gestaltet sind usw. Die Transformation der abstrakten Modelle in ein Implementationsmodell benötigt die Berücksichtigung umfangreicher Erkenntnisse der Software-Ergonomie und ist ein komplexer Arbeitsschritt, der in der Regel nicht auf Anhieb optimal gelingt.

7. Usability Inspection (Usability Untersuchung):

Eine Untersuchung der Usability - mit herkömmlichen Methoden, wie sie an anderen Orten ausführlich beschrieben werden, z.B. (Nielsen 1993) - wird an zwei unterschiedlichen Stellen im Prozess durchgeführt, nämlich vor und nach den Implementationsaktivitäten (Konzentrische Entwicklung / Iterativer Architektorentwurf). Constantine und Lockwood betonen den gezielten Einsatz von Benutzern in der Requirementsphase und im Zusammenhang mit Evaluation. Sie empfehlen jedoch, die Benutzerinnen nur sehr begrenzt in die eigentlichen Design-Aktivitäten zu involvieren.

8. Prioritized Concentric Construction (Konzentrische Entwicklung):

Konzentrische Entwicklung ist ein Prozess zur schichtweisen Entwicklung lauffähiger Systeme. Durch die Verwendung von Use Cases kann der gesamte Implementationsprozess, inklusive der Releaseplanung, gesteuert werden. Use Cases definieren sinnvolle Einheiten, die nach und nach entwickelt und geliefert werden können. Hierzu müssen die Use Cases zunächst priorisiert werden. Kommt es im Laufe der Projektdauer zu einem Entwicklungsstopp, so ist die Chance gross, dass zumindest ein Teil des Systems für bestimmte Aufgaben funktionsfähig ist.

9. Architectural Iteration (Iterativer Architektorentwurf):

Diese Methode ist eng mit der konzentrischen Entwicklung gekoppelt. Der iterative Architektorentwurf ist eine Methode zur Erhaltung einer konsistenten Software-Architektur, während sukzessive neue Schichten an Funktionalität zum System hinzugefügt werden. Dies setzt voraus, dass die zu Beginn des Projekts gemachten grundlegende Annahmen, die zu einem Framework für die Systemarchitektur führen, in jeder Entwicklungsrunde überprüft und angepasst werden, z.B. die Partitionierung des Systems in Packages, die Code-Struktur, die Organisation der Datenbank, die Klassenhierarchien usw.

10. Operational Contextualization (Kontextanpassung):

Parallel zu allen Modellierungs- und Designaktivitäten wird das Design an den tatsächlichen Anwendungskontext angepasst. Systeme müssen nicht nur an die Aufgaben angepasst sein, die damit erledigt werden sollen, sondern auch an den jeweiligen Anwendungskontext. Die Aspekte des Anwendungskontextes, die mit hoher Wahrscheinlichkeit Auswirkungen auf die Designentscheidungen bezüglich der Benutzerschnittstelle haben werden, stellen das Operationelle Modell dar. Hierbei handelt es sich um eine Sammlung verschiedener operationeller und kontextbezogener Einflussfaktoren, die eine Rolle im Hinblick auf die Usability des Systems spielen können. Hierunter versteht man z.B. die Eigenschaften der Benutzerinnen und die Benutzerrollen, Aspekte der physischen Arbeitsumgebung, Eigenschaften und Beschränkungen der vorhandenen Hardware oder allgemeine bzw. spezifische Risikofaktoren im Zusammenhang mit der Anwendung des Systems. Je nach Anwendungskontext können die verschiedenen Aspekte der Benutzerschnittstelle unterschiedlich gewichtig werden (z.B. Antwortzeitverhalten, leichte Erlernbarkeit usw.).

11. Standards and Style Definition (Erstellung eines Styleguides):

Dass die beiden Aktivitäten Kontextanpassung und Styleguideerstellung so spät durchgeführt werden, ist ein Bruch mit vielen herkömmlichen Herangehensweisen und beruht auf folgender Überzeugung: „superior designs result by focusing initially on the nature and structure of the work and then later adapting the user interface to operating conditions and constraints (...)“ (Constantine und Lockwood 1999, S. 37). Vergleichbares gilt für eine Entwicklung von Styleguides, die in den Augen der Autoren unlogisch ist, bevor man nicht weiss, was eigentlich entwickelt und standardisiert werden muss.

12. Help System and Documentation (Erstellung des Hilfesystems und der Dokumentation):

Der Erstellung des Hilfesystems widmen Constantine und Lockwood eine eigene Aktivität. Sie argumentieren, dass ein schlechtes Hilfesystem selbst ein ganz passables System zu völliger ‚Unusability‘ verdammen kann.

Bei Usage-Centered Design handelt es sich eher um eine Neuinterpretation und Anreicherung eines objektorientierten Software-Entwicklungsprozesses als um einen eigenständigen Usability Engineering Prozess. Die Aktivitäten ‚Konzentrische Entwicklung‘ und ‚Iterativer Architektorentwurf‘ machen deutlich, wie eng Usage-Centered Design an den eigentlichen Software-Entwicklungsprozess gekoppelt ist. Damit ist eine wichtige Voraussetzung geschaffen, um auch Softwareentwicklerinnen mit ins ‚Usability-Boot‘ zu holen. Das sehr strukturierte Vorgehen und die explizite, systematische Trennung zwischen Anforderungserhebung und Lösungsentwurf auch auf der Ebene der Use Case Modellierung ist zwar aus Usability Engineering Blickwinkel nicht neu. Im Bereich der objektorientierten Software-Entwicklung ist diese Trennung jedoch ein

grundlegender Fortschritt, der nur dadurch erreicht werden kann, dass eine neue Form von Use Cases definiert wird («essential use cases»), die eine lösungsneutrale Spezifikation von Anforderungen erlaubt. Der Weg über weitgehende Abstrahierung der Konzepte und Kernideen, aus denen sich die Arbeit mit dem zu entwickelnden System zusammensetzt, steht in einem gewissen Gegensatz zu der Herangehensweise von Rosson und Carroll, die mit Szenarien arbeiten, die möglichst detailliert und konkret sein sollen. Es ist anzunehmen, dass Szenarien zur Dokumentation von Details und Besonderheiten besser geeignet sind, als abstrakte Use Cases. Um sich jedoch von dem Bisherigen zu trennen und wirklich innovative Lösungen entwickeln zu können, scheint die abstrakte Repräsentation der Anforderungen überlegen. Die Einschätzung drängt sich auf, dass es zum Teil wohl eine Kombination aus Abstraktion und Konkretheit braucht, um erfolgreich und benutzer- bzw. nutzungszentriert innovative Software entwickeln zu können.

3.2 Probleme bei der praktischen Umsetzung von Usability Engineering Prozessen

Bei der Umsetzung von benutzerzentrierten Entwicklungsprozessen treten häufig – unabhängig vom spezifisch gewählten Usability Engineering Prozess – Probleme auf. Diese hängen grösstenteils mit dem organisationalen Kontext zusammen, denn alle Veränderungen auf der Ebene von Entwicklungsvorgehen sind auch organisationaler Natur.

An dieser Stelle sollen die offensichtlichen Schwierigkeiten, die allgemein und im Zusammenhang mit den in Kap. 3.1 vorgestellten Prozessmodellen zu erwarten sind, zusammengefasst werden. Lösungsansätze zur Überwindung dieser Schwierigkeiten werden hier nur kurz skizziert. Diese fliessen in detaillierter Form in das Modell einer benutzerzentrierten Entwicklungsorganisation ein, das in Kap. 5 vorgestellt wird.

1. Usability Engineering - und Software Engineeringprozesse sind nicht integriert:

Sehr häufig wird in Organisationen neben dem Standardentwicklungsprozess ein eigenständiger Usability Engineering Prozess definiert, in dem alle benutzerzentrierten Aktivitäten abgehandelt werden sollen. Dies führt dazu, dass die Gemeinsamkeiten zwischen Usability Engineering und Software Engineering gar nicht erkannt werden. Stattdessen wird getrennt, was eigentlich eng zusammen gehört, mit in der Regel negativen Folgen für die Usability der zu entwickelnden Produkte.

Bei einer Trennung der Prozesse werden benutzerzentrierte Aktivitäten als zusätzlich und prozessfremd empfunden und nicht als selbstverständlicher Bestandteil der Entwicklungsaktivitäten. Dadurch läuft man Gefahr, dass bei einer veränderten Prioritätensetzung die benutzerzentrierten Aktivitäten wieder aufgegeben oder bei Ressourcenknappheit ‚wegrationalisiert‘ werden. (Wiebe 2000)

Die beiden Prozesse werden unter anderem deshalb oft getrennt, weil es als schwierig empfunden wird, die unterschiedlichen Konzepte zu vereinigen. Usability Engineering Prozesse sind hochgradig iterativ, während bei Standardentwicklungsprozessen oftmals zumindest auf dem Papier ein Wasserfallmodell abgebildet wird.

Ein anderes Problem ist die personelle Trennung zwischen denjenigen Projektbeteiligten, die sich um die Anforderungsspezifikation kümmern, denjenigen, die die Benutzerschnittstelle definieren bzw. designen und schliesslich den Ausführenden, die das System implementieren. All diese Personen sind häufig nicht Mitglieder eines Teams, sondern gehören oftmals sogar ganz unterschiedlichen Geschäftseinheiten an bzw. werden von extern hinzugezogen.

→ Um zu ermöglichen, dass Doppelspurigkeiten, Zielkonflikte und Kommunikationsdefizite auftreten, müssen die Aspekte des Usability Engineerings in den Standardentwicklungsprozess integriert werden. Innerhalb des Entwicklungsprojektes ist es sinnvoll, die Zuständigkeiten für die technische Realisierung und die Entwicklung von Usability personell zu trennen. Das gesamte Team muss jedoch gleichermaßen dafür in die Verantwortung genommen werden, dass ein Produkt unter vertretbarem Zeit- und Kostenaufwand, mit vertretbarer technischer Leistungsfähigkeit und guter Usability entwickelt wird. Werden diese Verantwortlichkeiten getrennt, sind Zielkonflikte vorprogrammiert und alle multifaktoriellen, systemübergreifenden Konzepte wie Usability, Datensicherheit oder Fehlertoleranz werden unterwandert.

2. Wenn niemand für die Usability des entstehenden Systems verantwortlich ist, werden benutzerzentrierte Aktivitäten nicht koordiniert bzw. nicht in ausreichendem Masse ausgeführt:

Die Frage ist jedoch, wer letztlich für die Usability eines Softwareproduktes zuständig ist. Wer ist verantwortlich für das Design der Benutzerschnittstelle, wer hat die Expertise und die Fähigkeiten, die funktionalen und qualitativen Anforderungen an das System zu erheben und in ein adäquates, konsistentes Design zu überführen? Tatsache ist, dass die Entwicklung von Usability eine Querschnitts-

aufgabe ist, die so viele Aspekte enthält, dass diese kaum von einer Person vollständig abgedeckt werden können. Dies bedeutet, dass die Verantwortlichkeit für die Usability des Endproduktes weit verteilt werden muss! Es darf nicht der Eindruck entstehen, dass durch Hinzuziehen einer einzelnen Person mit entsprechenden Usability Engineering Kenntnissen die Usability des Endproduktes gewährleistet ist. (Constantine und Lockwood 1999)

→ Neben den Designerinnen der Benutzerschnittstelle, müssen sowohl die Analysten der Geschäftsprozesse, die Requirementsexpertinnen, die Marketingfachleute als auch die Entwicklerinnen ihren Anteil zur Usability des Endproduktes beitragen. Eine zentrale Einheit von Usability Experten kann helfen, die notwendige Expertise innerhalb der Organisation bereitzustellen, zu pflegen und zu verbreiten. Durch eine weitgefassete Formulierung der benutzerzentrierten Aktivitäten sollten jedoch alle Projektbeteiligten ihren Aufgabenbereich bezüglich Benutzerzentrierung wahrnehmen können. Mindestens eine Person im Projekt sollte explizit für die Usability des Produktes zuständig sein.

3. Die Einführung von Praktiken und Methoden des Usability Engineerings in einer Organisation bedeuten immer einen organisationellen Lernprozess (Metzker 2001). Bevor erkannt werden kann, dass viele Softwareprodukte an mangelnder Usability erkranken, muss ein gewisses Mass an Verständnis und Wahrnehmung für das Thema Usability vorhanden sein. An Stellen, wo dieses Problembewusstsein vorhanden ist, mangelt es teilweise an Kenntnissen, wie die Usability von Softwareprodukten verbessert werden kann. Da es nur wenige erfahrene Expertinnen im Bereich der benutzerzentrierten Software-Entwicklung gibt, ist die sachgerechte, effektive und effiziente Durchführung von Usability Engineering Methoden nicht unbedingt gewährleistet und es ist für unerfahrene Mitarbeiterinnen schwierig, die vorhandenen Prozesse effizient an die jeweiligen Projektbedürfnisse anzupassen.

→ Um dem Dilemma zu entgehen, dass schlecht durchgeführte benutzerzentrierte Massnahmen die Akzeptanz dieser Methoden unterwandern und andererseits Know-how nur durch Erfahrungen gebildet werden kann, sollte zum einen die notwendige Expertise von aussen herangezogen werden, Know-how innerhalb der Organisation sollte systematisch gefördert werden und es ist zudem sinnvoll, eine Usability Infrastruktur zu errichten, die die benutzerzentrierten Aktivitäten unterstützt und effizienter macht (z.B. Bereitstellung von Schulungen/Schulungsmaterial, Methodenhandbüchern, Beratungsleistungen, Styleguides und Guidelines in gebrauchstauglicher Form, Usability Patterns usw.).

4. In vielen Entwicklungsorganisationen sind die Rahmenbedingungen der Software-Entwicklung wenig geeignet, um benutzerzentrierte Entwicklungsprozesse zu fördern. Die Zeit- und Kostenvorgaben für die Entwicklung sind in den letzten Jahren immer härter geworden. Der enge Zeitrahmen führt oft dazu, dass mit der Realisierung begonnen wird, bevor überhaupt die Anforderungen spezifiziert sind. Der grosse Druck, früh sichtbare Ergebnisse zu liefern, führt (vor allem bei strategisch wichtigen Projekten) zur frühen Implementation unausgereifter Projektideen, zu inkonsistenten oder unzureichenden Konzepten und zu qualitativ fragwürdiger Umsetzung der immer aufs Neue hinzukommenden Anforderungen. Dieser Trend hat sich eingestellt, obwohl die Sinnhaftigkeit einer systematischen Anforderungsspezifikation nicht angezweifelt wird. Benutzerzentrierte Aktivitäten müssen genauso wie das [Requirements Engineering](#) gegen diese Situation ankämpfen.

→ Benutzerzentrierte Aktivitäten müssen vor der Einführung auf ihre Effizienz und Effektivität hin geprüft werden. Die Einführung muss schrittweise stattfinden und es muss sichtbar gemacht werden, dass Usability Engineering Vorgehen die bis anhin übergeordneten Kosten- und Zeitziele nicht unterwandern. Zudem muss der Erfolg eines Entwicklungsprojektes zusätzlich daran gemessen werden, ob die Benutzer des entwickelten Systems tatsächlich effektiver, effizienter und befriedigender arbeiten können als vorher. Hierzu ist eine Erfolgskontrolle nach Einführung des Systems notwendig.

5. Die Einführung von benutzerzentrierten Vorgehensweisen stellt eine grosse Veränderung zu herkömmlichen Software-Entwicklungsprozessen dar. Dies kann nur realisiert werden, wenn vom Management die entsprechende Unterstützung gewährleistet wird. Das Management muss bereit sein, die notwendigen Mittel zu investieren, um eine Usability Infrastruktur zu schaffen. Zudem müssen klare Signale gesetzt werden, die der Usability der entwickelten Produkte eine adäquate Priorität beimessen, um bei Zielkonflikten ausgewogene Entscheidungen treffen zu können.

→ Das Management muss die Bedeutung von Usability als kritischer Erfolgsfaktor erkennen und aktiv für die Anliegen der benutzerzentrierten Software-Entwicklung einstehen.

3.3 Zusammenfassung

Bei genauer Betrachtung lässt sich feststellen, dass sich die vorgestellten Usability Engineering Vorgehen in ihren Grundsätzen kaum unterscheiden. Alle Prozesse verfolgen als grundsätzliche Ideen:

1. Die frühe Einbeziehung von Benutzerinnen (unterschiedlich ist allenfalls das Ausmass, in dem man die Benutzer aktiv in den Designprozess einbezieht);
2. Die Analyse der Benutzeraufgaben und des Nutzungskontextes als Basis für das Design der Benutzerschnittstelle;
3. Die iterative Entwicklung der Benutzerschnittstelle mit Hilfe von Prototyping und kontinuierlicher Evaluation.

Diese Prinzipien erscheinen für das Usability Engineering so generisch, dass sie praktisch von allen Unternehmen, die einen eigenen Usability Engineering Prozess eingeführt haben, verfolgt werden. Z.B. bei der Hiser Group (Bloomer und Wolfe 1996), bei SAP (Erleben und Gebauer 2000), bei Reuters (Garrison et al. 1996), bei IBM (IBM 1999, Vredenburg 1999) und vielen mehr.

Unterschiede zwischen den Prozessen gibt es in der Gewichtung der Designaktivitäten, die teilweise auf mehrere (Abstraktions-)Ebenen aufgeteilt werden. Auch in der Reichweite unterscheiden sich die Prozesse signifikant. Am schwächsten schneidet dabei das Contextual Design ab, das nur die Konzeptphase behandelt, nicht jedoch den konkreten Entwurf der Benutzerschnittstelle. Der grösste Unterschied ist jedoch, ob die Nähe zum Software-Entwicklungsprozess gesucht wird oder nicht. Der Usability Engineering Lifecycle orientiert sich explizit am objektorientierten Software-Engineering (OOSE). Noch ausgeprägter ist diese Annäherung an einen OOSE Prozess beim Usage-Centered Design. Hier ist vor allem der Formalisierungsgrad bemerkenswert. Für die Repräsentation der Aufgaben, die mit der neuen Applikation zu erledigen sind, werden sogenannte Essential Use Cases entwickelt, deren Formalisierungsgrad die in der OOSE üblicherweise verwendeten Use Cases sogar noch übertrifft. Im Gegensatz zum Usability Engineering Lifecycle und Usage-Centered Design sucht das Scenario-based Design die Integration mit dem Software-Entwicklungsprozess praktisch nicht.

Betrachtet man die Probleme, die im Zusammenhang mit der Einführung benutzerzentrierter Entwicklungsaktivitäten zu erwarten sind, so gewinnt der zuletzt genannte Unterschied der Prozesse an Bedeutung. Um die organisationellen Schwierigkeiten zu überwinden ist es wichtig, die konzeptionelle Nähe zwischen Usability Engineering und Software Engineering zu betonen. Die benutzerzentrierten Aktivitäten müssen so gestaltet und kommuniziert werden, dass sie nicht in einem Widerspruch zur vorhandenen Entwicklungskultur einer Organisation stehen. Die Anpassung der vorhandenen Usability Engineering Prozesse an den jeweiligen organisationellen Kontext ist daher eine zentrale Aufgabe bei der Einführung benutzerzentrierter Entwicklungsprozesse. Dass die Durchführung benutzerzentrierter Methoden ebenso wie andere Software-Entwicklungsmethoden eine entsprechende Qualifikation auf seiten der Mitarbeiterinnen verlangen, versteht sich nahezu von selbst.

Damit benutzerzentrierte Aktivitäten unter den Bedingungen immer kürzerer Entwicklungszeiten und knapper Ressourcen bestehen können, muss ihre Effizienz und Effektivität belegt oder zumindest plausibel gemacht werden. Begründbar innerhalb der Projektlogik werden alle benutzerzentrierten Vorgehen jedoch frühestens dann, wenn die Usability des Endproduktes als gleichwertiges Erfolgskriterium neben Kosten- und Zeiteinhaltung gewichtet wird.

Um den grossen Schritt von technikzentrierter hin zu benutzerzentrierter Software-Entwicklung aktiv voranzutreiben, müssen neben dem eigentlichen Entwicklungsprozess vor allem organisationelle Aspekte betrachtet werden. Dies bedeutet konkret die Förderung von Innovation und organisationellem Lernen in der Organisation und eine Anpassung der organisationellen Rahmenbedingungen, unter denen Software entwickelt wird.

4 Benutzerzentrierung als Innovation

Die Ausführungen in Kapitel 2 und 3 dieser Arbeit legen nahe, dass die mangelnde Usability von Software-Produkten nicht darauf zurückzuführen ist, dass keine geeigneten Mittel oder Methoden zur Verfügung stünden, um diese systematisch zu entwickeln. Die Frage scheint vielmehr zu sein, warum Usability Engineering Prozesse und Methoden nicht oder kaum zur Anwendung kommen. Betrachtet man Benutzerzentrierung als das, was sie für viele Entwicklungsorganisationen darstellt, nämlich als Innovation, kann möglicherweise die Innovationstheorie einige Hinweise geben, wie die Situation verbessert werden könnte. Dies soll im folgenden versucht werden:

Ob und in wie weit Organisationen bereit und in der Lage sind, Innovationen wahrzunehmen oder zu entwickeln und für ihre Zwecke nutzbar zu machen, hängt von vielerlei Faktoren ab, die einerseits die Organisation selbst, aber auch die jeweilige Innovation betreffen:

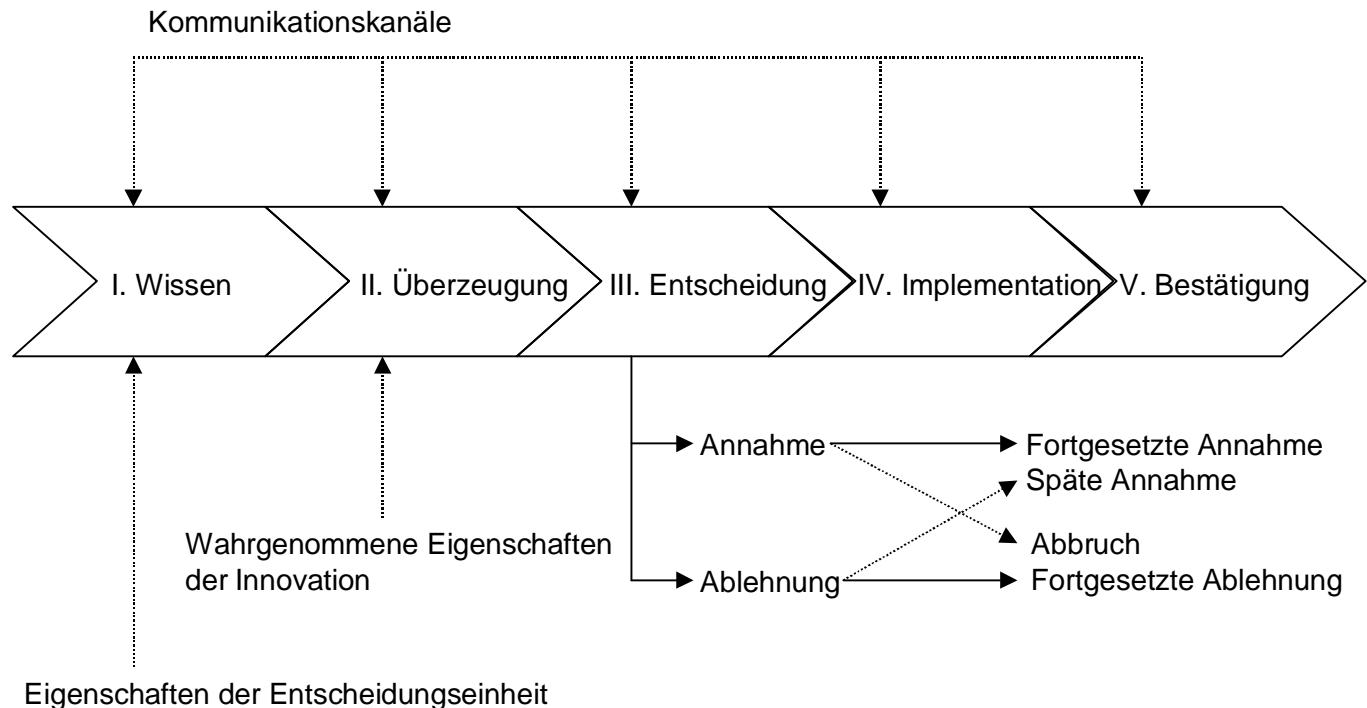
„Getting a new idea adopted, even when it has obvious advantages, is often very difficult. Many innovations require a lengthy period, often of many years, from the time they become available to the time they are widely adopted. Therefore, a common problem for many individuals and organizations is how to speed up the rate of *diffusion* of an innovation.“ (Rogers 1983, S. 24)

Obwohl gerade in Unternehmen Innovationsprozesse oft weniger vom Problem als vielmehr von einer vorhandenen Lösung getrieben werden, ist der Auslöser für die **Entwicklung** von Innovation in der Regel das Empfinden eines Problems oder eines Bedürfnisses, wie es sicherlich auch im Bereich der Software-Entwicklung der Fall war: Ende der 60er Jahre hat man bereits festgestellt, dass die vorhandenen Vorgehensweisen der Software-Entwicklung nicht erfolgreich auf die neuen, grösseren Projekte übertragen werden konnten. Es kam immer häufiger zu einer Übersteigerung des Kosten- und Zeitrahmens und zu vielen gescheiterten Entwicklungsprojekten, so dass man sogar von einer **Softwarekrise** sprach. Auch heute ist die Situation nicht wirklich gut. Laut einer Untersuchung aus dem Jahr 1999 wird nur ein Drittel der Software-Projekte zur Zufriedenheit des Kunden und des Anbieters durchgeführt. Ein weiteres Drittel liefert ein Produkt oder eine Dienstleistung, die in wesentlichen Teilen den Anforderungen des Kunden nicht entspricht und das letzte Drittel scheitert gar völlig und wird ohne Leistungserbringung abgebrochen (Schönwälder 2002, S. 4). Die zweite grosse Enttäuschung in der Software-Entwicklung war die Erkenntnis, dass die hohen Produktivitätserwartungen nach Einführung von Computerprogrammen, die den Alltag der Menschen erleichtern und unterstützen sollten, vielfach nicht erfüllt wurden (Landauer 1995). Diese Erfahrungen haben in den letzten 20 - 30 Jahren grosse Forschungsbemühungen im Bereich des Software-Engineerings und des Usability Engineerings ausgelöst, die unter anderem in den in Kapitel 2 und 3 dargestellten Prozessmodellen und Bewertungsmethoden vermarktet wurden. Die spannende Frage im Zusammenhang mit dieser Arbeit ist nun, wie und in welchem Umfang es gelingt, diese Innovationen im Zusammenhang mit dem Software-Entwicklungsprozess in den Organisationen anzunehmen, zu implementieren und in Zukunft vollständig zu integrieren.

Die tatsächliche Implementation einer Innovation in einem Unternehmen ist das Ergebnis eines Innovationsprozesses, der sich grob in drei Abschnitte unterteilen lässt: Initiation, Adoption und Implementation (Wynekoop und Senn 1992). Während der Initiationsphase werden die Probleme der Organisation definiert, nach Lösungsmöglichkeiten Ausschau gehalten und am Ende eine Innovation entwickelt oder ausgewählt. In der Adoptionsphase findet die Entscheidung statt, Ressourcen in die Innovation zu investieren. Die Implementation beinhaltet alle Aktionen, die nach der Entscheidung notwendig sind, um die Innovation effektiv zu benutzen und erfolgreich in die Organisation zu integrieren. Da es sich um Individuen handelt, welche Innovationen dann tatsächlich anwenden, spielt auch der individuelle Innovationsprozess innerhalb des organisationellen Innovationsprozesses eine Rolle. Welche Faktoren im gesamten Prozess erfolgskritisch sind, soll in den folgenden Abschnitten näher beleuchtet werden.

4.1 Der Innovationsprozess in Organisationen

Das Annehmen einer Innovation, also die Bereitschaft, eine Innovation in das Repertoire der verfügbaren Technologien, Prozesse oder Methoden aufzunehmen, ist das Ergebnis eines Entscheidungsprozesses, der mit der Wahrnehmung der Innovationsoption beginnt (egal, ob aus einem Problembewusstsein heraus oder weil sich die Option zufällig auftut) und mit der vollständigen Implementation der Innovation endet (so dass sie am Ende nicht mehr als solche existent ist). Dieser Prozess ist keineswegs immer völlig rational und hängt wiederum von vielen verschiedenen Faktoren ab. Zudem muss bei einem erfolgreichen Innovationsprozess immer sowohl die Ebene des Individuums als auch die der Organisation abgedeckt werden.



(nach Rogers 1983, S. 163)

Abbildung 10: Modell eines Innovationsprozesses in Organisationen

Die Phasen des Innovationsprozesses (nach Rogers 1983) (vgl. Abbildung 10):

- I In der Wissens-Phase des Innovationsprozesses geht es darum, überhaupt mit einer Innovation in Kontakt zu kommen. Diese Phase wird maßgeblich durch Eigenschaften der Individuen bzw. Organisationseinheiten geprägt, die die Innovationsentscheidung zu fällen haben. Kosmopolite Individuen oder weltoffene Organisationen werden häufiger auch ohne ein gezieltes Bedürfnis oder die Wahrnehmung von Problemen über Innovationen nachdenken, von denen sie über den hier wichtigsten Kommunikationskanal, nämlich überregionale oder globale (Massen-)Medien, erfahren. Die organisationalen Voraussetzungen, wie z.B. Bedürfnisse und Innovationsfreudigkeit, prägen die Offenheit einer Organisation gegenüber Innovationen ganz generell.
- II In der Überzeugungsphase spielen die wahrgenommenen Eigenschaften der Innovation eine grosse Rolle. Je überzeugender sich die Vorteile einer Innovation darstellen, desto positiver ist die Attitüde, die die Individuen oder die Organisation gegenüber der Innovation entwickeln. Entscheidend hierbei ist auch, als wie neu eine Innovation wahrgenommen wird und wie hoch die Unsicherheit ist, die damit assoziiert wird. Eine wohlwollende oder ablehnende Attitüde gegenüber einer Innovation bedeutet noch nicht deren tatsächliche Annahme oder Ablehnung. Allerdings besteht eine Tendenz, Verhalten und Attitüde in Übereinstimmung zu bringen. Bei der individuellen Bewertung der Innovation spielen interpersonelle Kommunikationskanäle eine wachsende Rolle, denn die Bewertung einer Innovation findet häufig auf der Basis oder in Abstimmung mit engen Kollegen statt.

- III In der Entscheidungsphase kommt es entweder zu einer Annahme oder zur Ablehnung der Innovation. Die Ablehnung wird entweder bewusst beschlossen (aktiv) oder äussert sich darin, dass eine Innovation nicht weiter verfolgt wird (passiv). Auch in dieser Phase sind die lokalen, interpersonellen Kommunikationskanäle entscheidend. Die Art der Innovations-Entscheidung ist zudem ausschlaggebend dafür, wie stabil bzw. von welcher Qualität die Entscheidung ist und hat Einfluss darauf, wie überzeugt die Mitglieder der Organisation von der Innovation sind und mit welchem Elan die Innovation umgesetzt wird. Bei Innovations-Entscheidungen in Organisationen lassen sich drei Typen differenzieren:⁶
1. Optionale Innovations-Entscheidung: Ein Individuum entscheidet sich unabhängig von der Entscheidung anderer Mitglieder der Organisation zur Annahme oder Ablehnung einer Innovation.
 2. Kollektive Innovations-Entscheidung: Eine Organisation entscheidet sich im Konsens mit allen Mitgliedern für oder gegen eine Innovation.
 3. Autoritäts-Innovations-Entscheidung: Relativ wenige Individuen innerhalb einer Organisation, die Macht, Status und/oder technische Expertise besitzen, entscheiden sich für oder gegen eine Innovation.
- IV In der Implementationsphase geht es darum, die Innovation zu modifizieren und an die spezifischen Gegebenheiten einer Organisation anzupassen (re-invention). Je besser eine Innovation in die Kultur und das Wertegefüge eingepasst werden kann, desto höher ist die Adoptionsrate und der nachhaltige Erfolg einer Innovation. In dieser Phase spielt der Innovationsprozess einzelner Individuen eine grosse Rolle: Um eine Innovation implementieren zu können, müssen Individuen vier Arten von Wissen besitzen: Wahrnehmungswissen (awareness knowledge) – um überhaupt von der Existenz der Innovation zu wissen, Bewertungswissen (evaluation knowledge) – um zu einer eigenständigen Entscheidung über die Annahme der Innovation kommen zu können, Nutzungswissen (use knowledge) – um zu wissen, wie die Innovation gewinnbringend genutzt werden kann und Prinzipienwissen (principles knowledge) – um die Prozeduren und unterlegten Konzepte der Innovation zu verstehen.
- V In der Bestätigungsphase wird eine angenommene Innovation entweder weitergeführt oder zurückgewiesen. Wenn es nicht gelingt, auch die letzten der beiden oben genannte Wissensarten bis dahin zu vermitteln, ist die Wahrscheinlichkeit, dass die Innovation in dieser Phase fehlschlägt, sehr gross.

4.2 Attribute von Innovationen

Ob und wie schnell eine Innovation von einer Organisation implementiert wird, hängt nicht zuletzt davon ab, welche Attribute die Innovation selbst aufweist. Mögliche Attribute von Innovationen werden im folgenden erläutert:

- Kompatibilität (compatibility): Wenn Innovationen als konsistent mit dem existierenden Wertesystem, vergangenen Erfahrungen, Bedürfnissen und kürzlich eingeführten Ideen wahrgenommen werden, so spricht man von Kompatibilität der Innovation. Je grösser die Kompatibilität ist, desto höher ist die Adoptionsrate einer Innovation.
- Relativer Vorteil (relative advantage): Besteht die Einschätzung, dass eine Innovation zu einem Wettbewerbsvorteil führt, ist ihre Umsetzung in Unternehmen leichter zu erreichen. Neben ökonomischen Vorteilen gibt es noch andere Aspekte, die als relative Vorteile empfunden werden, einer davon ist der Status-Aspekt.⁷ Je grösser der relative Vorteil einer Innovation empfunden wird, desto höher ist die Bereitschaft der Individuen bzw. der Organisation, die Innovation anzunehmen. Die Adoptionsrate steigt.
- Komplexität (complexity): Komplexität bezeichnet das Mass, in dem es als schwierig empfunden wird, eine Innovation zu verstehen oder zu handhaben. Je höher die Komplexität ist, desto geringer ist die Adoptionsrate.

⁶ Es sind auch bedingte Innovations-Entscheidungen möglich, bei denen die endgültige Entscheidung erst nach und nach auf der Basis von weiteren Informationen oder Erfahrungen gefällt wird.

⁷ Individuen, die als sehr innovativ gelten und immer ‚vorne mit dabei sind‘ (early adopters), geniessen oftmals ein hohes Ansehen innerhalb ihres sozialen Systems. Dies führt teilweise sogar dazu, dass es zu einer Art Überanpassung kommt und Innovationen angenommen werden, von denen Expertinnen der Meinung sind, dass sie eher abgelehnt werden sollten.

- Testbarkeit (trialability): Die Testbarkeit bezeichnet das Mass, in dem es möglich ist, eine Innovation in beschränktem Umfang vor der vollständigen Implementation auszuprobieren. Je höher die Testbarkeit einer Innovation ist, desto höher ist die Adoptionsrate.
- Überprüfbarkeit (observability): Die Überprüfbarkeit ist das Mass, in dem Auswirkungen und Ergebnisse einer Innovation für andere sichtbar und erkennbar sind. Je höher die Überprüfbarkeit ist, desto höher ist die Adoptionsrate.

Studien über US-amerikanische Farmer haben gezeigt, dass Innovationen, die den grössten ökonomischen Vorteil versprachen und als am wenigsten riskant eingeschätzt wurden, von den Farmern schneller angenommen wurden als andere Innovationen. Weniger relevant war es, wie komplex die Innovation war, wie gut deren Erfolge beobachtbar waren oder wie leicht eine Innovation zunächst getestet werden konnte. Am schnellsten wurden diejenigen Innovationen aufgegriffen, die am ehesten kompatibel waren mit dem Wertesystem der Farmer. (Rogers 1983)

4.3 Innovationsfreudigkeit einer Organisation

Organisationen sind soziale Systeme, die geschaffen wurden, um i.d.R. grossskalige Routineaufgaben durch ein Muster an regularisierten menschlichen Beziehungen zu bewältigen. Die Effizienz von Organisationen ist das Resultat einer erfolgreichen Zusammenführung menschlicher Bemühungen in einer Struktur, die sich durch gemeinsame, vordefinierte Ziele, definierte Rollen, Regeln und Vorschriften und zahlreiche informelle Muster auszeichnet (nach Rogers 1983).

Organisationen weisen Eigenschaften auf, die sie zum raschen und erfolgreichen Aufgreifen von Innovationsoptionen mehr oder weniger geeignet erscheinen lassen. Diese Faktoren beziehen sich sowohl auf die Individuen der Organisation als auch auf interne und externe Eigenschaften der Organisation selbst. In empirischen Studien konnte gezeigt werden, dass die Innovationsfreudigkeit einer Organisation deutlich positiv mit ihrer Grösse korreliert ist: je grösser eine Organisation ist, desto höher scheint ihre Innovationsfreudigkeit zu sein. Allerdings ist anzunehmen, dass die Grösse einer Organisation ein Art Surrogat ist für zahlreiche andere Messgrössen, die für die Innovationsfreudigkeit von Organisationen relevant sind, wie z.B. die verfügbaren Ressourcen, die Qualifikation der Mitarbeiterinnen oder die Organisationsstruktur. Im folgenden werden einige Faktoren aufgezählt, bei denen zumindest eine schwache Korrelation zur Innovationsfreudigkeit von Organisationen durch empirische Untersuchungen gezeigt werden konnte (Rogers 1983):

Faktoren, die positiv mit der Innovationsfreudigkeit einer Organisation korrelieren:

1. Attitüde gegenüber Veränderungen: Die Einstellung einer Organisation gegenüber Veränderungen ist eine sehr abstrakte Grösse. Letztlich handelt es um die Einstellung einzelner Mitglieder einer Organisation. Ist beispielsweise der Direktor eines Instituts oder der CEO einer Unternehmung sehr offen gegenüber Innovationen, so prägt dies die Attitüde der gesamten Organisation.
2. Komplexität: Das Mass, in dem Mitglieder einer Organisation im Besitz von Wissen und Expertise sind. Gemessen über das Mass an Spezialisierung und formale Ausbildung der Organisationsmitglieder.
3. Innere Vernetztheit: Mass, in dem die verschiedenen Einheiten einer Organisation über interpersonelle Netzwerke miteinander verbunden sind.
4. Organisationeller Überfluss: Mass, in dem innerhalb der Organisation ungebundene Ressourcen zur Verfügung stehen.
5. Systemoffenheit: Mass, in dem die Mitglieder einer Organisation mit anderen Individuen verbunden sind, die ausserhalb der Organisation angesiedelt sind.

Faktoren, die negativ mit der Innovationsfreudigkeit einer Organisation korrelieren:

1. Zentralisierung: Mass, in dem Macht und Kontrolle innerhalb einer Organisation in den Händen von wenigen Individuen konzentriert sind.
2. Formalisierung: Mass, in dem eine Organisation die Einhaltung von Regeln und Prozeduren betont.

In Abbildung 11 sind die Faktoren zusammengefasst.

Individuelle (Führungs-)Eigenschaften

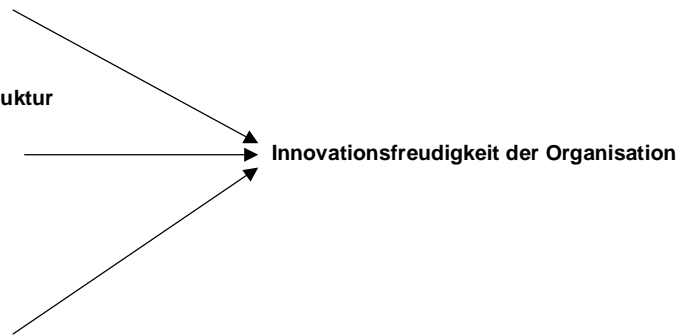
1. Attitüde gegenüber Veränderungen (+)

Interne Eigenschaften der Organisationsstruktur

1. Zentralisierung (-)
2. Komplexität (+)
3. Formalisierung (-)
4. Interne Vernetztheit (+)
5. Organisationeller Überschuss (+)
6. Grösse (+)

Externe Eigenschaften der Organisation

1. Systemoffenheit (+)



(nach Rogers 1983, S. 380)

Abbildung 11: Faktoren für die Innovationsfreudigkeit von Unternehmen

4.4 Innovationsfreudigkeit eines Individuums

Wie gesagt kommt es nicht nur darauf an, dass eine Innovation kollektiv angenommen wird. Die effektive Umsetzung einer Innovation ist vielmehr direkt davon abhängig, wie überzeugt die einzelnen Individuen von den Vorteilen durch die Anwendung der Innovation sind.

4.4.1 Motivation

Eine beliebte Massnahme zur Erhöhung der Innovationsfreudigkeit oder zur Steigerung der Risiko- und Leistungsbereitschaft der Mitarbeiterinnen in Unternehmen ist der Einsatz von sogenannten Incentives. Der Erfolg von Incentives beruht darauf, dass der relative Vorteil einer Innovation, der vom Individuum empfunden wird, steigt. Es lassen sich fünf verschiedene Aspekte von Anreizsystemen unterscheiden: (1) Entweder richten sich die Anreize an die Individuen oder an das gesamte System, (2) die Anreize können entweder positiv oder negativ sein, (3) sie können monetär oder nicht monetärer Art sein und (4) entweder sofort oder zeitverzögert ausgegeben werden. Der grösste Unterschied liegt jedoch darin, ob (5) der Anreiz für diejenigen Individuen geschaffen wird, die eine Innovation adoptieren (*adopters*) oder für solche, die deren Adoption verbreiten (*diffusers*). Anreizsysteme können zwar generell die Adoptionsrate von Innovationen erhöhen. Allerdings führen die Systeme, welche die ‚adopters‘ ansprechen, auch dazu, dass möglicherweise andere Individuen eine Innovation annehmen, als dies ansonsten der Fall wäre. Dadurch kann es zu einer geringeren Qualität der Innovations-Entscheidungen kommen, wodurch die Wirksamkeit der Innovation limitiert werden kann. Einen anderen Aspekt bezüglich Incentives führt Sprenger an. In seinem Buch „Mythos Motivation“ untersucht er die herkömmlichen Führungsstrategien zur Förderung der Motivation von Mitarbeiterinnen. Er stellt fest, dass Belohnungssysteme in hohem Masse dazu führen, die intrinsische Motivation der Mitarbeiter zu zerstören. Daraus schliesst er, dass Belohnung mit Sicherheit nicht das beste Mittel zur Leistungssteigerung ist (Sprenger 1998).

Wendet man diese Erkenntnisse auf die Förderung benutzerzentrierter Software-Entwicklungsprozesse an, so lässt sich daraus schliessen, dass Anreizsysteme allenfalls dazu dienen können, die grundsätzliche Werthaltung des Managements und das entsprechende Commitment für benutzerzentrierte Vorgehen zu transportieren. Auf keinen Fall kann man jedoch davon ausgehen, dass alleine über entsprechende Anreizsysteme nachhaltig die Durchführung benutzerzentrierter Prozesse gefördert werden kann. Stattdessen ist die ausreichende Kommunikation der Ziele und Vorteile durch die Anwendung benutzerzentrierter Methoden, aber auch der zu erwartenden Probleme mit der Innovation für die Motivation der Mitarbeiter unabkömmlich.

4.4.2 Wahrnehmung von Rollen

Im Zusammenhang mit Innovationsprozessen in Organisationen spricht man vom sogenannten Innovationschampion, der sich der Innovation verschrieben hat und sie enthusiastisch unterstützt und fördert. Eine andere wichtige Rolle spielt die Sponsorin, die Ressourcen für den Implementationsprozess der Innovation verfügbar macht und die notwendigen Gelder, Personal und andere organisationelle Ressourcen sichert. Champion und Sponsorin können in einer Person vereinigt sein.

Empirische Untersuchungen im Zusammenhang mit der Einführung eines CASE Tools haben gezeigt, dass das Vorhandensein von Personen, die diese Rollen einnehmen, einen grossen Einfluss auf den Erfolg der Einführung hat (Wynekoop und Senn 1992). Ausserdem ist bedeutsam, welche Position Champions und Sponsoren innehaben. Je höher die Sponsoren in der Hierarchie der Organisation angesiedelt sind und je aktiver ihre Unterstützung der Innovation ist, desto grösser sind in der Regel die zur Verfügung gestellten Ressourcen, z.B. für Ausbildung bezüglich der Innovation. Der Erfolg einer Innovation ist deutlich positiv mit dem Rang des Champions in der Hierarchie der Organisation korreliert. Auch im Usability Engineering ist die Bedeutung eines Champions bekannt (siehe z.B. Billingsley 1995, Mayhew 1999).

Daraus könnte man nun schliessen, dass ein Top-Down Ansatz, bei dem das oberste Management die Einführung einer Innovation befiehlt und entsprechend aktiv verfolgt und unterstützt, zwangsläufig zum Erfolg führt. Dies ist jedoch nicht der Fall, denn gerade ein Top-Down Ansatz, der die Mitarbeiterinnen dazu zwingt, womöglich gegen ihre Überzeugung eine Innovation anzunehmen, senkt die individuelle Akzeptanz. Vollständige Freiwilligkeit im Gebrauch von neuen Technologien und das komplette Zurückziehen des obersten Managements bezüglich der Implementation kann jedoch ebenfalls dazu führen, dass die Innovation nicht den gewünschten Erfolg erbringt.

4.5 Zusammenfassung

Man kann nicht davon ausgehen, dass eine Innovation – egal wie wünschenswert sie für die Organisation oder Teile der Organisation auch sein mag – sich selbst verkauft. Eine Implementation ist dann erfolgreich, wenn sie sowohl die **organisationelle** als auch die **individuelle** Ebene umfasst, die letztlich für die Qualität der Umsetzung entscheidend ist.

Auf organisationeller Ebene wird Innovation am ehesten implementiert, wenn eine klare Stellungnahme des obersten Managements erfolgt, dass die Umsetzung der Innovation erwartet wird. Ausserdem bedarf es einer Modifikation der organisationellen Prozesse (z.B. eine veränderte Leistungsbeurteilung der Mitarbeiterinnen oder zusätzliche erforderliche Projekt-Artefakte). Konkrete Massnahmen, um die Implementation einer Innovation in der Organisation zu fördern, beinhalten folgende Punkte:

- Verfügbarkeit substanzieller Ressourcen;
- Anpassung der Innovation zur leichteren Integration in die vorhandenen Prozesse und Vereinbarkeit mit der herrschenden Organisationskultur;
- Direkte Kommunikation zwischen Management, implementierendem Personal und den potentiellen Anwendern;
- Sorgfältige und klare Kommunikation der Vor- und Nachteile der Innovation;
- Sichtbares Management Commitment;
- Einen Champion / Sponsor aus dem oberen Management;
- Schaffung eines innovations- und experimentierfreudigen Arbeitsklimas, das den Mitarbeiterinnen erlaubt, Innovationen auszuprobieren und ihre Erfahrungen damit zu sammeln.

Eine erfolgreiche organisationelle Implementation geht nicht immer einher mit der erfolgreichen Implementation auf der Ebene der einzelnen Individuen. So kann es vorkommen, dass trotz vollständiger (z.B. erzwungener) Anwendung der Innovation innerhalb der Organisation das Ergebnis schlecht ist oder dass trotz grosser Begeisterung einiger Mitglieder (z.B. mangels Ressourcen) die Verbreitung innerhalb der Organisation ausbleibt. Dabei darf nicht vergessen werden, dass Aktivitäten, die zum Erfolg einer Innovationsimplementation auf der einen Ebene beitragen, dem Erfolg auf der anderen Ebene abträglich sein können.

Erfolgreiche individuelle Implementation der Innovation wird am ehesten erzielt durch:

- Freiwilligkeit in der Verwendung einer Innovation;
- Ausbildung von Mitarbeiterinnen hinsichtlich Kenntnisnahme, Evaluation, Prinzipien und Anwendung der Innovation;
- Förderung der Wahrnehmung eines relativen Vorteils durch die Annahme der Innovation;
- Reduktion der wahrgenommenen Komplexität der Innovation durch schrittweise Einführung.

5 Die benutzerzentrierte Entwicklungsorganisation – Ein Reifegradmodell

Wie in Kapitel 3 beschrieben, gibt es inzwischen nicht nur zahlreiche Einzelmethoden des Usability Engineerings (wie beispielsweise Benutzeranalysen, Kontext- und Aufgabenanalysen, diverse Prototyping und Usability Testverfahren usw.), sondern auch umfassende Prozessmodelle. Diese sollen ermöglichen, systematisch «Usability» zu entwickeln, in dem während der gesamten Entwicklungsdauer benutzerzentrierte Aktivitäten durchgeführt werden.

Man sollte meinen, dass nun, da das nötige Werkzeug vorhanden ist, der Entwicklung von Produkten mit hoher Usability nichts mehr im Wege steht. Die Realität des Usability Engineerings sieht jedoch anders aus. Erfolgsberichte beschreiben in der Regel Einzelfälle und selbst Deborah Mayhew muss eingestehen, dass es ihr erst kürzlich zum ersten Mal gelang, ihren Usability Engineering Lifecycle in adaptierter Version gesamthaft anzuwenden. Von dem Ziel, **organisationsweit**, d.h. in allen Projekten, benutzerzentriert Software zu entwickeln, sind wir also noch Lichtjahre entfernt. Carlshamre und Rantzer übertiteln ihre Erfahrungen mit der Delta Method pointiert mit „Failure of a Success Story“, da trotz ausgiebiger Schulung, Verbreitung und sogar Anwendung des Delta Prozesses weniger als 2% (1.4%) aller Entwicklungsprozesse bei der Firma Ericsson (in deren Auftrag die Delta Method entwickelt wurde) beeinflusst werden konnten (Carlshamre und Rantzer 2001). Zahlen von anderen Organisationen liegen meines Wissens nicht vor, es ist allerdings davon auszugehen, dass auch in anderen Unternehmen die Quote der Anwendung von Usability Engineering Methoden oder gar vollständigen Prozessen kaum höher ist. Im Bewertungsmaßstab des Usability *Maturity Models* (Earthy 1999) oder des DATEch-Prüfbausteins: Usability-Engineering-Prozess (Deutsche Akkreditierungsstelle Technik e.V. 2001a) würden Unternehmen wie Ericsson sehr gut abschneiden, denn es gibt einen definierten Usability Engineering Prozess, der die Entwicklungsprojekte bewiesenermaßen dazu befähigt, gebrauchstaugliche Produkte zu entwickeln. Trotz aller erfolgreichen Bemühungen ist jedoch auch Ericsson keine wirklich benutzerzentrierte Entwicklungsorganisation geworden. Auf die Frage, warum das so ist und wie man diese Situation überwinden könnte, gibt es bisher keine befriedigenden Antworten. In einer Untersuchung über die Einbettung von Usability Engineering in die organisationellen Prozesse, die Kultur und die Produktlandschaft und die Bedeutung von Usability Daten in unternehmensweiten Entscheidungsprozessen, kommen Rosenbaum et al. zu wenig neuen oder gar überraschenden Ergebnissen. Bei diesem sogenannten ‚Strategic Usability‘ konnten zwar bestimmte Korrelationen festgestellt werden zwischen der Verwendung einzelner Usability Engineering Methoden und deren geschätzter strategischer Effizienz. Die Ergebnisse dieser sehr umfangreichen Studie, an der sich über 130 Usability Profis beteiligt haben, reichen jedoch über einige Gemeinplätze nicht hinaus, wie z.B. dass Usability Experten die Sprache des Business sprechen sollen, genügend Kenntnisse über die technischen Abläufe erarbeiten sollten, um mit den Softwareingenieuren diskutieren zu können und dass Usability Testing als eine Methode eingeschätzt wird, die die grösste strategische Bedeutung besitzt (Rosenbaum et al. 1998).

Diese Arbeit folgt der Idee, dass für die erfolgreiche Implementation eines benutzerzentrierten Entwicklungsprozesses in einer Organisation, die organisationellen Rahmenbedingungen mit gleicher Aufmerksamkeit betrachtet werden müssen, wie der Entwicklungsprozess selbst. Dies ist in der Vergangenheit zu wenig beachtet worden und hat dazu geführt, dass die kulturelle Entwicklung in den Organisationen nicht mit dem Fortschritt in den Usability Engineering Methoden und Prozessen mithalten konnte.

Um Anhaltspunkte zu schaffen, wo man ansetzen muss oder wo eine Organisation auf ihrem Weg zur Benutzerzentrierung noch Mängel aufweist, wird in dieser Arbeit – angelehnt an die Idee eines benutzerzentrierten Prozessmodells – ein Modell einer idealtypischen benutzerzentrierten Software-Entwicklungsorganisation entworfen. Das Modell basiert auf Konzepten der vorhandenen Software-Engineering und Usability Engineering Prozesse, den theoretischen Überlegungen zur Innovationsverbreitung in Unternehmen, Ideen zu Know-how-Entwicklung und -Transfer und zahlreichen Tipps und Tricks, die zu diesem Thema vor allem in der Usability Engineering Literatur zu finden sind.

Insgesamt werden 7 Dimensionen beschrieben, die für die Benutzerzentriertheit einer Entwicklungsorganisation als relevant erachtet werden. Die vorgestellten Dimensionen entsprechen formal in etwa den Prozesskategorien eines Prozessreifegradmodells, wie beispielsweise SPICE oder CMM. Zu jeder Dimension werden in den folgenden Kapiteln Forderungen formuliert, die an eine Entwicklungsorganisation gestellt werden sollten. Die Forderungen entstammen aus der einschlägigen Usability Literatur oder sind das Ergebnis eigener Überlegungen und Erfahrungen. Die Erfüllung der Forderungen wird als Voraussetzung

dafür erachtet, dass Benutzerzentrierung erfolgreich und nachhaltig Einzug in alle Entwicklungsprozesse halten kann. Bei den Forderungen, die formal auf der Stufe der *base practices* bzw. *generic practices* von SPICE angesiedelt sind, handelt es sich um Eigenschaften und nicht um Vorgehensweisen. Die Forderungen werden später in Merkmale benutzerzentrierter Organisationen uminterpretiert.

Da es unrealistisch ist davon auszugehen, dass Organisationen von heute auf morgen alle Forderungen erfüllen können, werden diese in einem zweiten Schritt 3 Reifestufen zugeordnet. Auf Stufe 1 sind all diejenigen Forderungen zu erfüllen, die sich entweder in besonderem Masse als Initialschritte eignen oder die eine Voraussetzung darstellen, um darauf aufbauend die Benutzerzentriertheit der Organisation zu erhöhen. Auf Stufe 2 hat die Organisation einen Status erreicht, auf dem Benutzerzentrierung in den für die Software-Entwicklung wichtigsten Bereichen verankert ist. Auf Stufe 3 ist Benutzerzentrierung organisationsweit umgesetzt und die Organisation ist in der Lage, ihre Erfahrungen zu einer systematischen Verbesserung der Vorgehensweisen und Prozesse einzusetzen.

Das Organisationsmodell in Kombination mit dem Ratingverfahren soll Usability Greenhorns, Usability Consultants und Entwicklungs-Verantwortlichen und Managerinnen innerhalb einer Organisation dabei behilflich sein, festzustellen:

- was sie alles berücksichtigen sollten,
- wo Schwachstellen vorhanden sind,
- welche Massnahmen ergriffen werden sollten, um den Stand der Benutzerzentrierung innerhalb der Organisation zu verbessern.

Das Modell soll als Basis dienen, um konkrete Handlungsanleitungen auszuarbeiten und systematisch das Thema Benutzerzentrierung innerhalb der Organisation zu fördern. So soll es gelingen, dem Ziel gebrauchstauglicher Produkte mit allen daraus resultierenden Vorteilen näher zu kommen. Ein Nebeneffekt dieses Zieles ist es, Innovation sowohl auf Produkt- als auch auf Prozessebene zu fördern, den Software-Entwicklungsprozess besser beherrschbar zu machen und eine Unternehmenskultur zu schaffen, die die Lernfähigkeit und Innovationsfreudigkeit der Mitarbeiter fördert.

5.1 Dimensionen einer benutzerzentrierten Entwicklungsorganisation

Wie in dieser Arbeit bereits verschiedentlich beschrieben wurde, spielen neben dem eigentlichen Software-Entwicklungsprozess zahlreiche andere Faktoren für die erfolgreiche benutzerzentrierte Software-Entwicklung eine Rolle. Ziel des hier entwickelten Modells einer benutzerzentrierten Organisation ist es, diese Faktoren thematisch zu bündeln und in einzelnen Bewertungsdimensionen zusammenzufassen, um sie so für eine systematische Evaluation und kontinuierliche Verbesserung zugänglich zu machen (vgl. Abbildung 12):

1. Ein für die Organisation passendes Entwicklungsprozess-Modell, um gebrauchstaugliche Produkte zu entwickeln und den Entwicklungsprozess zu kontrollieren.
2. Ein Portfolio an geeigneten Methoden und Werkzeugen, um die im Prozessmodell definierten Ziele zu erreichen.
3. Geeignete Mitarbeiter zur Ausübung der anfallenden Aufgaben im Bereich der Usability-Infrastruktur und der Durchführung von benutzerzentrierten Aktivitäten.
4. Eine Organisationsstruktur, in der Usability Engineering einen angemessenen Platz hat und die einen effizienten und effektiven Einsatz aller Entwicklungsressourcen ermöglicht.
5. Managementaktivitäten und Kultur, die die erfolgreiche Implementation von Benutzerzentrierung auf individueller und organisationaler Ebene fördern.
6. Ein Schulungswesen und Wissensmanagement, das zu einer Verbreitung, Erhaltung und Weiterentwicklung des Usability Know-hows im Unternehmen beiträgt
7. Ausreichende Ressourcen, um das Thema Usability im Unternehmen zu verankern und weiter zu entwickeln.

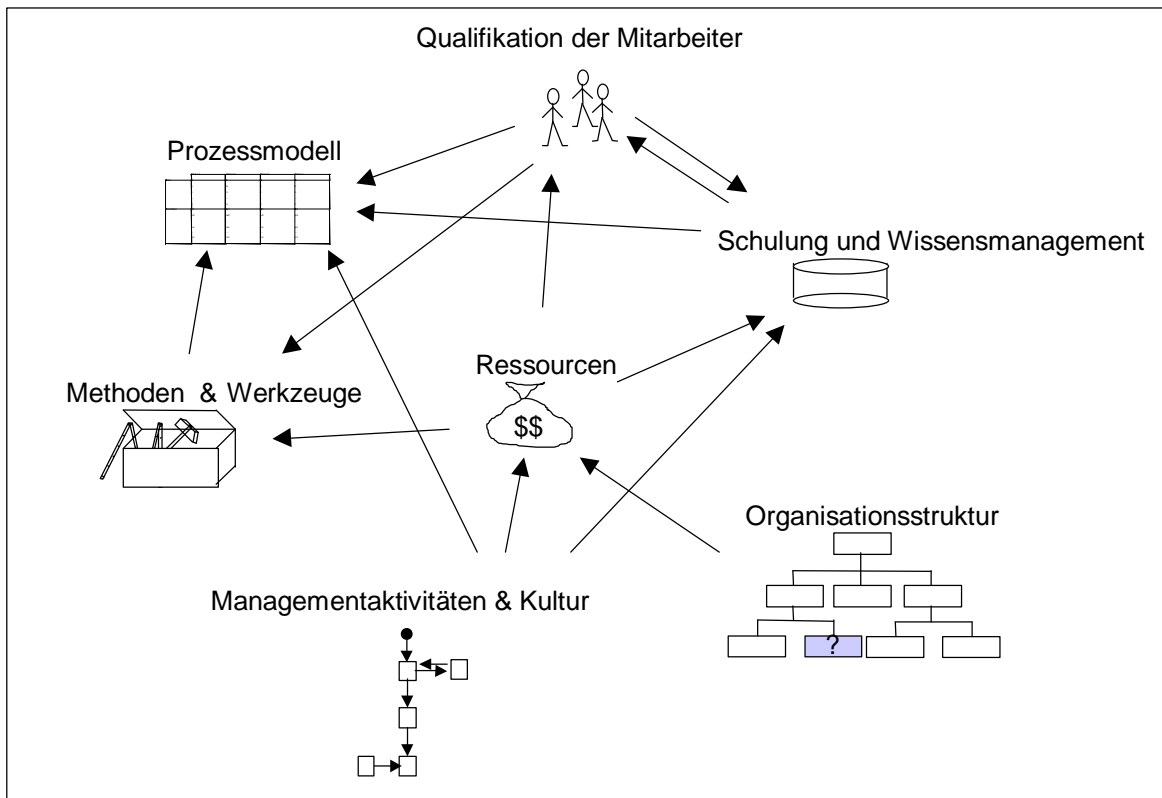


Abbildung 12: Dimensionen von Benutzerzentrierung

Im folgenden sollen die einzelnen Dimensionen näher beleuchtet werden und Anforderungen formuliert werden, die in einer idealtypischen benutzerzentrierten Organisation erfüllt sein müssen.

5.1.1 Ein Prozessmodell für benutzerzentrierte Software-Entwicklung

Ziel dieser Arbeit ist es nicht, das 1001ste Prozessmodell zu entwerfen. Es soll vielmehr dargestellt werden, was einen guten Entwicklungsprozess ausmacht und nach welchen Kriterien er bewertet werden muss, um seine Tauglichkeit zur erfolgreichen organisationsweiten Entwicklung von benutzerfreundlichen Produkten zu bewerten.

→ Das Prozessmodell muss Usability Engineering und Software-Engineering Prozesse vereinen.

Auf der Ebene des Endprodukts tragen sowohl die Aspekte des Usability Engineerings als auch die des Software-Engineerings gleichermaßen zum Erfolg eines Software-Entwicklungsprojektes bei. Bisher wird jedoch wenig Wert auf eine Vereinigung der verschiedenen Ansätze gelegt, was neben den meist sträflich getrennten Ausbildungswegen und gegenseitiger Ignoranz der beiden Disziplinen (Faulkner und Culwin 2000) vermutlich auch damit zu tun hat, dass nach wie vor ein Konflikt zwischen den beiden Ansätzen wahrgenommen wird, der verbunden ist mit dem Kampf um Ressourcen. Weder die Usability Community noch die Riege der Software Entwicklerinnen ist bereit, eine gemeinsame Basis in Form von gleichwertig nebeneinander existierenden Zielsetzungen und Prozessaktivitäten zu formulieren – auch wenn diese bei genauer Betrachtung eine deutliche Schnittmenge aufweisen. Bei Verzicht auf eine Vereinigung werden Usability Belange jedoch auch weiterhin als **zusätzlicher** Aufwand empfunden werden, denn selbst ein schlechtes Produkt ist immerhin noch ein Produkt auch ohne jegliche Usability Engineering Aktivitäten. Erst wenn sowohl die technischen als auch die benutzerzentrierten Ziele gleichwertig nebeneinander stehen, ist es möglich, auf konstruktive Art und Weise, die vielen *Trade-offs*, die dabei anfallen, tatsächlich anzugehen und unter Einbeziehung aller wichtigen Aspekte (Design-)Entscheidungen zu fällen. Hierzu ist ein integriertes Vorgehen von Usability und Software-Engineering notwendig, für das es zumindest aus der Schule der objektorientierten Software-Entwicklung und des *Requirements Engineerings* einige vielversprechende Ansätze gibt (Hudson 2001), (Ferré 2001), (Leffingwell und Widrig 2000).

→ Das Prozessmodell muss Iteration unterstützen.

Ein Grund, der vielfach für die fehlende Integration von Usability Engineering und Software-Engineering Prozessen genannt wird, ist die scheinbare Unvereinbarkeit: Usability Engineering ist hochgradig iterativ, Software Engineering Prozesse werden oftmals linear und wasserfallartig beschrieben. Auf der formalen Ebene dürfte dies kein Hinderungsgrund sein, da auch Usability Engineering Prozesse – vermutlich mangels einer geeigneteren Darstellungsweise – oft sequentiell abgebildet werden (vgl. Z.B. das Usability *Maturity* Model (Earthy 1998).⁸ Auch bei der Ausführung besteht nicht eigentlich ein Konflikt bezüglich Linearität und Iteration, da bei grossen Entwicklungsprojekten diese Linearität auch im Software-Engineering nicht durchzuhalten ist. Neuere Entwicklungsprozesse, wie beispielsweise der Rational Unified Process (RUP) (Probasco 2000), Extreme Programming (XP) oder die Dynamic Systems Development Method (DSDM) (DSDM-Consortium) sind hoch iterativ und haben keinerlei Ähnlichkeiten mehr mit einem Wasserfallprozess. Ob und inwieweit ein Prozessmodell Iteration unterstützt, hängt neben der Art wie es sich präsentiert auch von der Formulierung der Aktivitäten und den Anforderungen zur Qualitätskontrolle ab. Wird im Prozess die Ablieferung bestimmter Ergebnisse zu einem gewissen Zeitpunkt verlangt, auch wenn eigentlich klar ist, dass diese im Laufe des Prozesses weiterentwickelt werden müssten, wird Iteration nicht nur nicht gefordert, sondern an wichtigen Punkten auch unterwandert, da viel Aufwand betrieben werden muss, um auf unsicherer Basis Ergebnisse zu produzieren. Bei wirklich iterativen Prozessen müssen Arbeitsprodukte kontinuierlich weiterentwickelt werden. Qualitätskontrolle muss sich daher eher auf der planerischen und auf der Prozessebene abspielen als auf der Ebene von Artefakten.

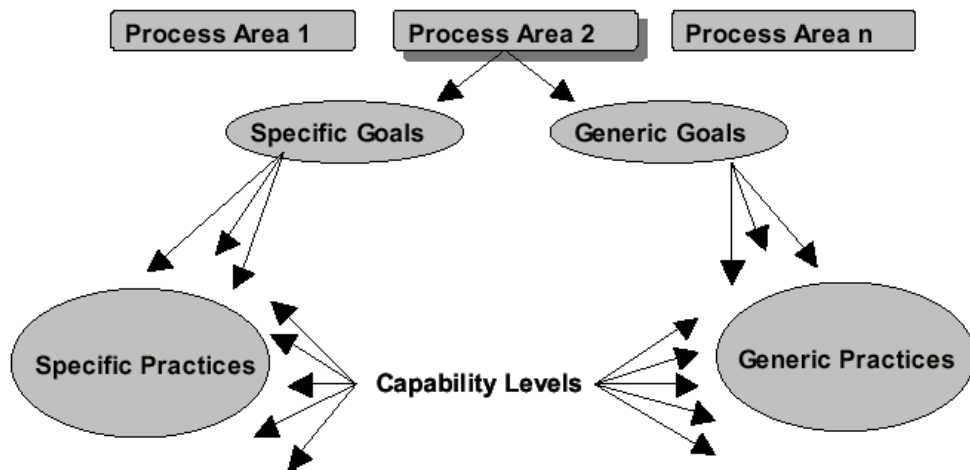
→ Es müssen Prozesskategorien sowie dazugehörige Aktivitäten und quantifizierbare Ziele definiert sein, die auch den Benutzeraspekt beinhalten.

Typischerweise – und da sind sich praktisch alle in Kap. 2.2 beschriebenen Vorgehensweisen einig – werden in den Prozessmodellen Ziele definiert, die mittels bestimmter Aktivitäten erreicht werden sollen. Das Usability Maturity Model (UMM) definiert Ziele und Aktivitäten im Hinblick auf benutzerzentrierte Vorgehensweisen. Beispielsweise wird das Ziel formuliert, in allen Teilen der Organisation, die mit Systemvertrieb, -Konzeption, -Entwicklung und -Support zu tun haben, den Fokus auf die Anliegen der Stakeholder und Benutzerinnen zu legen. Als weiteres Ziel wird die Spezifikation aller benutzerzentrierten Aktivitäten und deren Einbindung in den Systementwicklungsprozess (bzw. system lifecycle process) und das Unternehmen genannt. Die Ziele müssen sich an den Bedürfnissen und Geschäftszielen der jeweiligen Organisation orientieren. Um die Zielerreichung überprüfen zu können, sollten dezidierte, nach Möglichkeit quantifizierbare Ziele formuliert werden. Z.B. „In allen Projekten mit mehr als 100 geplanten Benutzern werden Methoden zur Kontext- und Aufgabenanalyse durchgeführt“ oder „In Erstprojekten ohne Vorkenntnisse über den Anwendungskontext werden mindestens 25% der veranschlagten Ressourcen für die Anforderungsspezifikation verwendet“.

Darüber hinaus ist es sinnvoll, bestimmte Prozesskategorien zu definieren, denen die Ziele und Aktivitäten zugeordnet werden. Dies ist bedeutend, um später Analysen durchführen zu können, auf deren Basis der Prozess optimiert werden kann. Im Rahmen des *Capability Maturity* Models (CMM) wurden sogenannte ‚Process Areas‘ definiert. Diese stellen eine Gruppierung von Aktivitäten dar, die durchgeführt werden müssen, um gewisse Ziele zu erreichen (vgl. Abbildung 13).

Als Prozesskategorien können beispielsweise «Projektmanagement», «Anwendungsentwicklung», «Configuration Management» und «Qualitätssicherung» definiert werden, wie beispielsweise im Prozessmodell von UBS Schweiz (UBS AG 1999) oder «Usability Qualitätsziele», «Benutzerbeteiligung», «Personenqualifizierung», «Anforderungsentwicklung» usw., wie im DATech Prüfbaustein: Usability-Engineering-Prozess vorgeschlagen. Ziele und Aktivitäten sollten fortlaufend überprüft und gegebenenfalls modifiziert werden, sofern neue Erkenntnisse oder Erfahrungen ein Verbesserungspotential aufzeigen.

⁸ Dass es zumindest formal auch in nicht iterativen Prozessmodellen des Software-Engineerings möglich ist, die Abschnitte zu ergänzen, die das Usability Engineering in den verschiedenen Phasen der Entwicklung vorsieht, wurde bereits am Beispiel des V-Modells gezeigt (Pelka 2000). Immerhin bewirkt die Einbettung von benutzerzentrierten Aktivitäten in ein Prozessmodell, dass die wichtigsten Fragen hinsichtlich der Benutzer und ihrer Bedürfnisse gestellt werden und dieser Aspekt auch in einem technizienten Umfeld nicht vollkommen übersehen wird. Eine explizite Benutzerzentrierung und eine Optimierung der gesamten Organisation diesbezüglich ist durch alleinige Anpassung des Prozessmodells jedoch nicht zu erreichen.



(SEI 2000)

Abbildung 13: Zusammenhang zwischen Prozesskategorien, Zielen und Aktivitäten

Um eine zufriedenstellende Usability des Endproduktes zu erreichen, muss «Benutzerzentrierung» nach Möglichkeit eine eigene Prozesskategorie darstellen. Mindestens muss Benutzerzentrierung jedoch explizit innerhalb einer vorhandenen Prozesskategorie (z.B. der Qualitätssicherung) aufgeführt sein. Ausserdem müssen entsprechende Aktivitäten und (Usability Engineering) Ziele formuliert werden.

→ **Die Aktivitäten müssen methodenneutral formuliert sein.**

Um alle am Entwicklungsprojekt Beteiligten gleichermassen anzusprechen, ist es wichtig, nicht vorzuschreiben, mit welchen spezifischen Methoden die Aktivitäten bearbeitet werden müssen. Jede Disziplin bringt ihre eigenen Methoden mit und vor allem Entwickler möchten sich nicht vorschreiben lassen, **wie** sie etwas zu tun haben: „(,) if for once we did a user analysis on software engineers (..) we would understand that they don't like to be told *how* to do things. They are trained to solve problems, and they are usually amazingly skilled at just that.“ (Carlshamre und Rantzer 2001, S. 37)

Da jedoch davon auszugehen ist, dass auch diverse Aufgaben eines benutzerzentrierten Entwicklungsprojektes nicht ohne methodische Unterstützung erfolgreich zu bearbeiten sind, ist es notwendig, dass bei Bedarf auf etablierte Methoden oder Tools und ein entsprechendes Consulting zurückgegriffen werden kann. Sinnvollerweise werden auf der Ebene des Prozessmodells die durchzuführenden Aktivitäten mit den entsprechenden Methoden und Tools im Sinne von Vorschlägen verbunden.

→ **Verantwortlichkeiten im Prozess müssen über ein Rollenkonzept definiert werden, das die Benutzerzentrierung explizit abdeckt.**

Es ist seit langem üblich, für die wichtigsten Aspekte bei der Software-Entwicklung (z.B. Projektplanung und -leitung, Geschäftsprozessanalyse, Anforderungsanalyse, Design usw.) Fachpersonen zu benennen, deren Verantwortung darin liegt, für eine qualifizierte Ausführung der anfallenden Aufgaben zu sorgen. Sogenannte Rollenkonzepte in benutzerzentrierten Entwicklungsprozessen müssen neben den technischen und organisatorischen Rollen auch eine Rolle abdecken, die mit der Durchführung von benutzerzentrierten Aktivitäten betraut ist. Z.B. könnte die Rolle der Usability Verantwortlichen geschaffen werden, die für die Koordination und adäquate Durchführung von Usability Engineering Aufgaben verantwortlich ist.

→ **Der Prozess muss den kompletten Lifecycle abdecken.**

Software-Entwicklungsprozesse beginnen häufig mit dem konkreten Auftrag an ein oder mehrere Entwicklungsteams und enden mit der Ausbreitung eines Softwareproduktes. Es wäre jedoch sinnvoller, bestimmte Vorarbeiten, wie z.B. das Abstecken des Funktionalitätsumfangs des zu entwickelnden Produktes und ggf. eine Releaseplanung und Voruntersuchungen zu Benutzergruppen usw. explizit in den Prozess mit einzubeziehen. Das Ende eines Projektes sollte auch nicht durch die Ausbreitung des Produktes, sondern frühestens nach dem 2. Release gesehen werden. Dies ist auch deshalb wichtig, da im Falle von Usability der eigentliche Erfolg nicht vor ersten realen Nutzungserfahrungen zu beurteilen ist. Dieser **Produkt-Lifecycle** sollte komplett durch das Prozessmodell abgedeckt sein.

→ Das Prozessmodell muss skalierbar und modifizierbar sein.

Kein prototypischer Entwicklungsprozess ist für eine reale Organisation in jeder Situation und über längere Zeit hinweg adäquat und umsetzbar. Weder die Prozessmodelle des Software Engineerings, wie sie in CMM oder ISO 15504 vorgestellt werden, noch diejenigen des Usability Engineerings (UMM, DATech Prüfbaustein: Usability-Engineering-Prozess, Usability Engineering Lifecycle usw.) sind dazu gedacht, ohne Anpassungen von einer Organisation übernommen zu werden. Es ist zwar sinnvoll, ein generisches ideales Prozessmodell als eine Art Schablone und zur Orientierung für alle Projektverantwortlichen vorliegen zu haben. Diese Schablone muss jedoch ständig optimiert und von jedem Projekt entsprechend seiner Grösse und Komplexität neu instanziiert und angepasst werden: „While it might seem tempting to specify an ideal usability process and require clients to match that process, such an approach is not practical, is not in the best interests of the corporation, and does not match our belief that there is no ‚best‘ way to achieve usability. No single process accommodates the different cultures, knowledge, and development constraints. If we were to enforce one process for all clients, developers would simply ignore us.“ (Wixon und Comstock 1994, S. 154f)

→ Die Präsentation des Prozessmodells muss gebrauchstauglich sein.

Um Anpassungen an einem Prozessmodell vornehmen zu können, müssen die Verantwortlichen in der Lage sein, alle Zusammenhänge innerhalb des Prozessmodells zu verstehen und Auswirkungen von Modifikationen auf den Gesamtprozess zu erkennen. Jeder Projektleiter muss die in jedem Prozess vorhandenen ‚Musts‘ von den eher nebenläufigeren Aktivitäten unterscheiden können, um eine möglichst effiziente Prozessanpassung vornehmen zu können. Neben der entsprechenden Ausbildung bezüglich Projektmanagement und Entwicklungsprozess, bedeutet dies auch eine Anforderung an das Prozessmodell selbst: Das Prozessmodell muss sich in einer Form präsentieren, die es ermöglicht, sich schnell einen Überblick über den Gesamtprozess und die Bedeutung und Zusammenhänge der einzelnen Prozessabschnitte zu verschaffen, um die notwendigen Modifikationen gezielt vornehmen zu können.

Nicht jede Art der Prozessdarstellung ist gleichermassen gebrauchstauglich. Zum einen muss ein benutzerzentrierter Prozess ein iteratives Vorgehen fördern, zum anderen muss ersichtlich sein, wie die einzelnen Teile des Prozesses ineinander greifen. Die am besten erfassbare Form eines komplexen Prozesses scheint dessen Aufspaltung in übersichtliche Module und deren Darstellung entlang einer Zeitachse. Diese Darstellung läuft jedoch häufig der Iterativität des Prozesses entgegen.

Bei der Untersuchung der bisher diskutierten Prozessmodelle lassen sich 3 Varianten erkennen, bei denen die Integration von Software- und Usability Engineering Prozessen grundsätzlich vorstellbar ist:

- (1) In ein Phasen-Prozessmodell à la UMM werden alle Software- und Usability Engineering Aktivitäten integriert, wobei dann auch diese z.T. sequentieller erscheinen, als sie in der Realität ausgeführt werden können;
- (2) Man entwirft Prozesskategorien wie in der ISO 15504, die die einzelnen Schichten eines Software- und Usability Engineering Prozesses darstellen. Innerhalb dieser Schichten können zugehörige Aktivitäten definiert werden, die über ein zeitliches Kontinuum hinweg ausgeführt werden. Aktivitäten des Usability Engineerings bzw. der Benutzereinbeziehung in einen Entwicklungsprozess können eine oder mehrere Schichten im Prozessmodell darstellen;
- (3) Man schafft eine Art Matrixstruktur, indem Prozessabschnitte definiert werden, denen Aktivitäten der einzelnen Prozesskategorien zugewiesen werden. Die Kategorien bleiben erhalten, um die verschiedenen Prozessdimensionen sichtbar zu machen, sind aber zeitlich eingeordnet (ein Beispiel hierfür ist die Präsentation von RUP).

Variante (3) bietet den Vorteil, dass sie sich für eine gut verständliche und abstrahierbare Darstellungsweise eignet, mit der es möglich ist, sowohl einen Überblick über den Gesamtprojektablauf zu erhalten, als auch den zeitlichen Überlappungen und Iterationen gerecht zu werden.

→ Das Dokumentenhandling muss effizient und für den Entwicklungsprozess effektiv sein.

Qualitätssicherungsmaßnahmen sind sehr häufig dokumentenorientiert. D.h. die Einhaltung von Prozessvorgaben wird anhand der produzierten Ergebnisse beurteilt. Dies führt dazu, dass sehr viel Energie für die Erstellung dieser Ergebnisse verwendet wird, die vor allem der Kontrolle dienen, oft jedoch nicht dazu tauglich sind, tatsächlich damit zu arbeiten. Für kleinere Entwicklungsprojekte bedeutet dies den administrativen Overkill. Doch auch in grossen Projekten droht die Gefahr, dass eine kontinuierliche Aktualisierung der Dokumente nicht stattfindet und der Aufwand für die Erstellung der Dokumente von anderen, im Hinblick auf die Entwicklungsaufgaben wichtigeren Aktivitäten abhält. Wenn nun aufgrund der Anforderungen des Usability Engineerings zusätzliche Dokumente erstellt werden müssen, ist dies der Effizienz des Prozesses eher abträglich. Es besteht also der Bedarf, Mittel und Wege zu finden, wie die Ergebnisse eines Entwicklungsprozesses einerseits auf ein notwendiges Mass reduziert werden können und diese andererseits so zu

gestalten, dass sie tatsächlich als Arbeitswerkzeug tauglich sind. Im Zusammenhang mit Usabilityaspekten sind Ansätze hierfür z.B. die Dokumentation von Anforderungen mit Hilfe von Requirement Management Tools oder in Form von evolutionären oder – noch besser – kommentierten Wegwerfprototypen. Eine der noch offenen Fragestellungen diesbezüglich ist z.B., wie man spezifizierte Use Cases am besten dokumentiert, so dass sie eine direkte Transformation in ein entsprechendes Design optimal unterstützen.⁹

Bezüglich der Dokumentation ist im Software-Engineering seit längerem ein Trend hin zu ‚leichtgewichtigen‘ Prozessen, wie z.B. Extreme Programming (XP), festzustellen, deren Vorgehensweise durchaus vereinbar zu sein scheinen mit benutzerzentrierten Aktivitäten (Constantine 2001). Allerdings laufen gerade diese Leichtgewichte der Software-Entwicklung Gefahr, dass eine kohärente Struktur der Benutzerschnittstelle auf der Strecke bleibt (Schaffer 2001).

→ **Die Projektdokumentation sollte die systematische Aufarbeitung von usabilityrelevantem Wissen erleichtern.**

Es ist unrealistisch davon auszugehen, dass die Projektverantwortlichen während oder nach Abschluss eines Projektes willens und in der Lage sind, die Erfahrungen, die sie im Zusammenhang mit einem benutzerzentrierten Vorgehen gemacht haben, systematisch zu dokumentieren oder zu kommunizieren.

Dennoch sind diese Erfahrungen so wertvoll, dass Bemühungen gerechtfertigt sind, diese zu sammeln. Die Sammlung von Erfahrungen im Zusammenhang mit Usability Engineering sollte Aufgabe der zentralen Usability Einheit sein. Um diese Aufgabe zu erleichtern, ist die Schaffung eines speziellen Projektergebnisses sinnvoll, in dem alle Aspekte der Benutzerschnittstelle von der ersten Benutzer- und Anforderungsanalyse bis zum endgültigen GUI-Design bzw. tatsächlichen Produkt abgelegt werden sollen, im Sinne eines Produkt Styleguides bei (Mayhew 1999, Kap. 14: Style Guide Development).¹⁰

Diese Anforderung steht in einem gewissen Konflikt mit der vorhergehenden Anforderung, der an dieser Stelle nicht vollständig aufgelöst werden kann. Zwei Aspekte sind diesbezüglich zu berücksichtigen: Die Dokumentationsanforderungen an ein Entwicklungsprojekt müssen der Maxime „So wenig wie möglich, aber so viel wie nötig“ folgen und sollten zudem durch geeignete Ablagemöglichkeiten mit gemeinsamem Zugriff und einheitlicher Struktur unterstützt werden, so dass sie auch nach Abschluss eines Projektes noch sinnvoll auswertbar sind. So lange jedoch nicht gewährleistet werden kann, dass sie sinnvoll ausgewertet wird, sollte auf zusätzliche Anforderungen an die Dokumentation verzichtet werden.

→ **Das Prozessmodell muss eine Usability Evaluation beinhalten.**

Jedes seriöse Software-Entwicklungsmodell enthält Massnahmen zur Qualitätsprüfung. Trotz des engen Zusammenhangs zwischen gutem Prozess und gutem Produkt ist es notwendig, auch die Usability eines Produktes zu ermitteln. Die Evaluation ist auch im Hinblick auf die Verbreitung des Usabilitygedankens innerhalb der Organisation ein zentrales Element. Denn letztlich dient sie einerseits zur Bewusstmachung des Problems (sofern die Organisation noch nicht sehr fortgeschritten auf ihrem Weg zur benutzerzentrierten Software-Entwicklung ist), andererseits ist sie bei erfolgter Usability Engineering Praxis eine Demonstration der Potenz benutzerzentrierter Vorgehensweisen. Ausserdem dient nicht zuletzt die Usability Evaluation des Endproduktes dazu, den Entwicklungsprozess zu optimieren und Herangehensweisen herauszufiltern, die möglicherweise im vorhandenen Kontext nicht zum gewünschten Erfolg führen. Bei benutzerzentrierten Entwicklungsprozessen gehört das Einholen von Benutzerfeedback zum Standardablauf und Designvorschläge der verschiedensten Ausprägung werden kontinuierlich evaluiert. Aber auch Organisationen, die noch nicht so ‚reif‘ sind bezüglich ihrer Benutzerzentrierung sollten zumindest am Ende eines Entwicklungsprojektes eine Usability Evaluation durchführen und zwar zu einem Zeitpunkt, wo noch die Möglichkeit gegeben ist, die aufgedeckten Usabilitymängel zu beseitigen.

⁹ Use Cases sind sowohl ein Hilfsmittel des klassischen Software-Engineerings als auch des Usability Engineerings. Dieser Begriff wird jedoch für eine derart weite Palette an verschiedenen Modellierungen verwendet und tatsächlich herrscht noch ziemlicher Klärungsbedarf, welche Darstellung denn nun die benutzerzentrierte Entwicklung von Software und die Schaffung innovativer Softwarelösungen am besten unterstützt. Vgl. hierzu auch die Verwendung von «essential use cases» im Usage-Centered Design, die in Kap. 3.1.5 beschrieben wird.

¹⁰ Achtung: Der Begriff Styleguide ist in sehr vielen Unternehmen bereits für ein Dokument reserviert, das ein konsistenteres Design von Benutzerschnittstellen erlauben soll. Um Missverständnisse zu vermeiden, sollte für ein umfassendes Dokument, in dem alle Ergebnisse auf dem Weg zu einem spezifischen Benutzerinterface abgelegt sind, eher der Begriff «Benutzerschnittstellen Spezifikation» oder ähnliches verwendet werden.

5.1.2 Das Angebot geeigneter Usability Engineering Methoden und Werkzeuge

Gerade im Bereich des Usability Engineerings gibt es zahlreiche Vorgehensmodelle und Methoden, die nicht immer dazu einladen, diese auch tatsächlich auszuprobieren. Zum Teil sind die Prozesse schwer überblickbar (das beste Beispiel hierfür ist vermutlich das Buch „Contextual Design“ von Beyer und Holtzblatt 1998) und ausserdem präsentieren sie sich so, dass sie kaum ohne das Hinzuziehen von teuren Consultants oder Expertinnen durchführbar erscheinen, was wiederum zu einer sog. ‚Intimidation barrier‘ innerhalb der Organisation führen kann (Nielsen 2002, Nielsen 1994a).

Wie für alle Innovationen, ist auch für die Verbreitung von Usability Engineering Methoden erfolgskritisch, dass sie sich gut in das Gefüge der Organisation und ihrer Prozesse einpassen. Stehen die ‚neuen‘ Methoden zu sehr mit diesen im Widerspruch, muss eine Anpassung stattfinden, wobei zu klären ist, ob die Randbedingungen oder die Methoden angepasst werden müssen. Grundprinzipien des Usability Engineerings, wie z.B. die frühe Einbeziehung von Benutzern, frühes und kontinuierliches Testen mit Benutzerinnen, integriertes und iteratives Design (Gould et al. 1991) dürfen dabei natürlich nicht aufgegeben werden. Es ist jedoch abzuwägen, ob für die Erstellung von Benutzerprofilen tatsächlich ethnologische Studien durchgeführt werden müssen oder ob eine einfachere Vorgehensweise ausreichend ist.

→ Die Entwicklungsmethoden und -Tools müssen bewährt und an die speziellen Gegebenheiten der Organisation angepasst sein.

Um eine effiziente und effektive Anwendung von Usability Engineering Methoden zu ermöglichen, sollten diese zunächst in einem kleinen Rahmen von Fachpersonen angewendet werden, um sie dann an die jeweiligen spezifischen Gegebenheiten der Organisation anpassen zu können. Auf der Basis der so gewonnenen Erfahrungen können sogenannte *best practices* für die Organisation definiert werden, die dann für die Entwicklungsprojekte zur Verfügung gestellt werden können, wobei klar zu kommunizieren ist, welche Methoden dazu geeignet sind, von allen durchgeführt zu werden und für welche es eher Usability-expertinnen benötigt.

Ein anderer wichtiger Aspekt bezüglich Methoden ist noch, um welche Art von Entwicklungsorganisation es sich handelt. Grudin unterscheidet sinnvollerweise drei Arten von organisationellem Kontext für die Software-Entwicklung, die sich massgeblich auf die Beziehung zwischen Entwicklern und Benutzerinnen auswirken und daher auch grossen Einfluss auf die bereitzustellenden Methoden haben (Grudin 1996):

1. Entwicklung auf Vertragsbasis
2. In-house Entwicklung
3. Entwicklung von kommerziellen Produkten oder Paketen

Die Methoden der Benutzereinbeziehung müssen den jeweiligen Gegebenheiten angepasst werden. In Organisationen, die In-house entwickeln, kommen beispielsweise Methoden wie das aus Skandinavien stammende Participatory Design (Schuler und Namioka 1993) oder das bei IBM entwickelte Joint Application Development (JAD) (Wood und Silver 1995) mit Erfolg zur Anwendung, während in Organisationen, die kommerzielle Produkte herstellen, zunächst die Frage geklärt werden muss, was überhaupt entwickelt werden soll. Hier sind dann Prozesse, wie z.B. das Contextual Design (Beyer und Holtzblatt 1998), eher angemessen.

Sollen Entwicklungstools zum Einsatz kommen (z.B. Requirement Management oder Prototyping Tools), so müssen diese ebenfalls vorher in kleinerem Rahmen auf ihre Tauglichkeit geprüft werden, bevor die Organisation sich womöglich Lizenzen und Beratung für ein Tool leistet, das für die tatsächlich gegebenen Bedingungen innerhalb der Organisation gar nicht angemessen ist. Und auch hier gilt wieder, dass Usability Expertinnen ihre eigenen Pillen schlucken müssen: Die angebotenen Hilfsmittel, wie z.B. Styleguides und Guidelines, die Entwickler befähigen sollen, ergonomische Benutzerschnittstellen zu entwickeln, müssen selbst eine ausreichende Usability aufweisen, um sinnvoll eingesetzt werden zu können (Carter 1999).

→ Das Methodenangebot muss geeignet sein, um alle benutzerzentrierten Aktivitäten unterstützen zu können (Vollständigkeit).

Teilweise wird die Meinung vertreten, dass ja bis anhin selbst die grundlegendsten Prinzipien des Usability Engineerings nicht berücksichtigt werden und daher die offensive Politik des „discount Usability Engineering“ Ansatzes betrieben werden sollte (Nielsen 1994a). Das idealtypische Prozessmodell der Organisation sollte sich jedoch nicht mit „discount“ Lösungen zufrieden geben. In dem Masse, wie es gelingt, benutzerzentrierte Aktivitäten im Standard-Entwicklungsprozess einer Organisation zu verankern, müssen auch die entsprechenden Methoden und Werkzeuge vorhanden sein, um diese Aktivitäten effektiv ausführen zu können. Im Laufe eines kontinuierlichen Lernprozesses muss also das Methodenangebot gemeinsam mit dem Prozess wachsen.

→ **Die Methodenauswahl und die Methoden selbst müssen an das jeweilige Projekt anpassbar sein.**

Um die Anwendung von Usability Engineering Methoden für alle Arten von Projekten zu ermöglichen, müssen die Methoden für jedes Projekt angepasst werden können. Hierbei kann der Usability Engineering Lifecycle von D. Mayhew als Vorbild genommen werden, der sowohl für die einzelnen Methoden als auch für den gesamten Prozess Abkürzungen (Shortcuts) anbietet (Mayhew 1999).

Wichtig bei der Anpassung ist es, dass trotz Abkürzungen zentrale Fragen der benutzerzentrierten Software-Entwicklung nicht vergessen gehen. Um den Entwicklungsprojekten die am besten geeignete Auswahl und Ausprägung der jeweiligen Methoden anbieten zu können, braucht es daher entweder erfahrene und versierte Usability-Fachleute, die ein solches ‚Menü‘ zusammenstellen können, oder aber für besonders fortgeschrittene Organisationen sogar die entsprechenden Tools, wie z.B. GUIDE oder ProUse, die in Kap. 7.2 beschrieben werden.

5.1.3 Die passende Organisationsstruktur

Organisationen unterscheiden sich stark in Grösse und Struktur sowie in ihrem Entwicklungskontext (reine Entwicklungsorganisationen oder In-house Entwicklung, s.o.). Es wäre daher falsch, an dieser Stelle eine klare Vorgabe zu machen, wo eine Usability Engineering Einheit (im folgenden UE-Einheit genannt) innerhalb einer Organisation angesiedelt sein sollte.¹¹ Allerdings ist es wichtig, bestimmte strukturelle Rahmenbedingungen für eine effektive Benutzerzentrierung zu schaffen. Neben der Ansiedlung im Organigramm sind dabei auch noch folgende Fragen von Bedeutung: wie ist die UE-Einheit aufgebaut (zentralisiert oder verteilt)?, wie arbeiten die UE-Expertinnen in Projekten mit (ins Team integriert oder als Consultants)? und wem muss der Usability Engineer berichten? Viele grössere Firmen tendieren dazu, die Organisationsformen zu mischen, um die Komplexität bei der Organisation und Koordination der Aufgaben für tausende von Mitarbeiterinnen zu reduzieren (Borgholm und Halskov Madsen 1999).

→ **Die UE-Einheit einer ‚In-house‘ entwickelnden Organisation muss so platziert sein, dass sie sowohl IT initiierte als auch Business initiierte Entwicklungsprojekte unterstützen kann.**

Bei benutzerzentrierten Aktivitäten handelt es sich Querschnittsaufgaben, die sowohl IT- als auch Businessfragen einbeziehen müssen. Es kann daher von Vorteil sein, wenn die UE-Einheit keinem dieser beiden Bereiche direkt unterstellt und daher auch nicht besonders verpflichtet ist, sondern eher eine Stabsfunktion einnimmt. Grundsätzlich ist jedoch auch im In-house Entwicklungskontext die Ansiedlung in der Nähe der Entwicklungseinheit sinnvoll. Die räumliche und inhaltliche Nähe zu den Entwicklern erleichtert die Zusammenarbeit und unterstützt die Kooperation zwischen diesen Bereichen. Dies ist eine wichtige Voraussetzung für die effektive Entwicklung von Benutzerschnittstellen mit hoher Usability.

In jedem Fall muss jedoch ermöglicht werden, dass die benutzerzentrierten Aktivitäten beginnen können, bevor die eigentliche Entwicklung ins Spiel kommt. Die Spezifikation der funktionalen Anforderungen an das System sollte durch eine Usability Expertin unterstützt werden und bereits am Anfang des Prozesses sollte die weitere Benutzerbeteiligung in die Wege geleitet werden. Mit einem Team von 15 Entwicklern im Rücken, die gierig ihre ersten Codezeilen schreiben wollen, werden diese Vorarbeiten mit hoher Wahrscheinlichkeit vernachlässigt.

→ **Die UE-Einheit soll ein zentrales Standbein haben.**

Obwohl die Integration eines UE-Experten in ein Entwicklungsteam gewisse Vorteile mit sich bringt, ist es wichtig, eine zentrale Einheit aufzubauen, die sich um die Entwicklung und Weiterentwicklung von Know-how im Zusammenhang mit Benutzerzentrierung kümmert. Zu den Aufgaben einer solchen Einheit gehört auch das Schaffen von gewissen Standards, z.B. in Form von Guidelines und Styleguides. Doch nicht nur für die Erledigung bestimmter ‚Back-Office‘ Arbeiten ist eine zentrale Einheit sinnvoll. Vielmehr wird damit eine Möglichkeit geschaffen, das Thema Usability sichtbarer zu machen. Viele Firmen, die ein Usability Labor eingerichtet haben, haben die Erfahrung gemacht, dass damit das Thema in der Organisation eine Art räumliche Identifikationsfläche bekommen hat. Und nicht zuletzt aus organisatorischen Gründen ist eine zentrale Einheit sinnvoll, da es nur so möglich ist, die vorhandenen UE-Kräfte zu koordinieren.

¹¹ Dieses Thema scheint bisher noch sehr wenig explizit betrachtet worden zu sein. Seit dem letzten Jahr gibt es eine Newsgroup zur Ansiedlung von Usability Einheiten in einer Organisation und auf der CHI 2001 in Seattle hat sich hierzu eine Special Interest Group (SIG) gebildet, die sich explizit mit diesen Fragen befasst hat:

<http://groups.yahoo.com/group/UsabilityInOrgChart/>

→ **Die UE-Expertin soll sowohl innerhalb der Projekte als auch als Consultant und Coach tätig sein können.**

Obwohl das Fernziel einer Organisation sein sollte, dass möglichst alle mit der Entwicklung von Software befassten Personen zumindest Basiskenntnisse in der Durchführung von benutzerzentrierten Aktivitäten aufweisen, wird immer Unterstützungsbedarf durch interne (oder auch externe) Experten vorhanden sein, die besonders komplexe oder aufwändige Methoden durchführen können oder Mitarbeiter bei der Durchführung coachen. Die Arbeit als Consultant oder Coach ist eine effiziente Möglichkeit, innerhalb der Organisation die Ressourcen der UE-Expertinnen zu nutzen. Diese Art von Einsatz bringt zudem den Vorteil mit sich, dass die UE-Experten Einblick in die verschiedensten Projekte erhalten und die Kenntnisse der benutzerzentrierten Software-Entwicklung effizienter verbreitet werden können. Die Nachteile eines Consultings (anstelle der Mitarbeit als Teammitglied) liegen darin, dass die Kommunikation schwieriger ist, die Consultants ihre Zeit auf verschiedene Aufgaben aufteilen müssen und dass eine Priorisierung aller anfallenden Bedürfnisse stattfinden muss, da keine permanente Unterstützung möglich ist.

Sind UE-Expertinnen dagegen integraler Bestandteil eines Projektteams, haben sie den Vorteil, dass sie ihre klare Rolle innerhalb des Projektes einnehmen können und für alle Belange des benutzerzentrierten Vorgehens hinzugezogen werden können. Zudem kennen sie dadurch sehr viel besser die ganz spezifischen Rahmenbedingungen eines Projektes und können diese eher z.B. bei der Entwicklung ihrer Designvorschläge berücksichtigen. Das Problem hierbei ist vor allem, dass die Verstrickung mit dem Projekt so gross ist, dass kein objektiver Blickwinkel mehr eingenommen werden kann und dadurch möglicherweise die Benutzerinteressen anderen Projektinteressen geopfert werden. In der Literatur finden sich zwei Aussagen, die diesen *Trade-off* auf den Punkt bringen:

„Integrating usability engineers into individual project teams increases immersion and attention to project-specific usability issues, BUT usability engineers may become so immersed that they miss opportunities for cross-project relationships.“ (Rosson und Carroll 2002, S. 344)

„(,) if the usability engineer has been too intimately involved with the design and development of a product, he or she may lose objectivity. He or she might begin negotiations for a more usable interface from a more compromised position, simply because of the intimate awareness of the development environment’s constraints,...“ (Mayhew und Bias 1994, S. 300)

Grundsätzlich ist es empfehlenswert, dass je nach verfügbaren Ressourcen und Projektbedeutung beide Arten von Projektarbeiten möglich sind.

→ **Ein multidisziplinäres Team aus Business-Verantwortlichem, UE-Expertin, Interface-Entwickler und Interface-Designerin soll die Entscheidungsbefugnis über die GUI-Entwicklung haben.**

Da es bei der Entwicklung von gebrauchstauglichen Benutzerschnittstellen oftmals darum geht, zwischen ansprechendem (oder auch Corporate) Design, vielfältigem Funktionsangebot, aufgabenorientierten Abläufen sowie technischer Machbarkeit und Performanz abzuwägen, ist es sinnvoll, dass Entscheidungen bezüglich der Ausgestaltung der Benutzerschnittstelle von einem multidisziplinären Team gefällt werden, das in der Lage ist, alle relevanten Aspekte hierzu zu bedenken und der Entscheidung zugrunde zu legen. In der Realität wird dies zunächst häufig nicht der Fall sein. Für den UE-Experten ist es jedoch wichtig, dass er direkt mit derjenigen Person verhandeln kann, die letztlich die Entscheidungsbefugnis hat, um die Interessen der Benutzerinnen wirkungsvoll vertreten zu können.

5.1.4 Managementaktivitäten und Organisationskultur als kritische Erfolgsfaktoren

Der Erfolg einer Innovation ist sowohl von der Verbreitung der Innovation innerhalb der Organisation als auch von der Annahme einer neuen Idee durch die einzelnen Mitglieder dieser Organisation abhängig. Wie z.B. in Kap. 4.4 dargelegt, ist es nicht unbedeutend, wie sich eine Innovation in einer Organisation verbreitet. Es sollte angestrebt werden, dass die einzelnen Mitglieder sich freiwillig für die Ausübung einer Innovation entscheiden, anstatt hierzu auf die eine oder andere Art gezwungen zu werden. Dies trifft mit Sicherheit auch im Falle der ‚Innovation Benutzerzentriertheit‘ zu, wo es in erster Linie darum geht, dass alle am Entwicklungsprojekt Beteiligten, die besondere Bedeutung der Benutzereinbeziehung erkennen und auch unterstützen (mit welchen Methoden auch immer), um einen bestmöglichen Erfolg zu erzielen. Um dies nachhaltig zu realisieren, ist es erstrebenswert, dass Usability zu einem selbstverständlichen Teil der Organisationskultur wird:

„It is important that this [usability has a role throughout the development cycle, S.H.] is achieved through voluntary ‚buy-in‘ by management rather than coercion through senior management directives, as the methods will be much more effective if applied willingly and thus begin to ‚sell themselves‘. For this reason, getting usability into the everyday culture of the organization is crucial (...).“ (Saunders und Arnfeld, S. 70)

In diesem Kapitel soll nun dargelegt werden, welche Managementaktivitäten auf einer Ebene über dem eigentlichen Entwicklungsprozess benötigt werden, um beides in ausreichendem Masse zu fördern und damit auch die Grundlage für die erfolgreiche Implementation eines benutzerzentrierten Software-Entwicklungsvorgehens zu schaffen.

→ **Die Innovationsfreudigkeit der Mitarbeiter muss gefördert werden.**

Aus der Beobachtung, dass bestimmte Organisationen innovationsfreudiger sind als andere, lassen sich einige Faktoren ableiten, die die grundsätzliche Bereitschaft fördern, sich auf neue Ideen einzulassen. Innerhalb der Organisation muss eine Kultur gepflegt werden, die offen ist für neue Ideen und Prozesse. Dies wird am besten dadurch erreicht, dass die Mitarbeiter dazu ermuntert werden, problemorientiert nach neuen Lösungsmöglichkeiten zu suchen und Neues auszuprobieren. Führende Mitglieder der Organisation besitzen dabei eine Vorbildrolle. Sie sind es, die die neuen Ideen in die entsprechenden Foren tragen müssen und die Mitarbeiterinnen dazu anhalten können, bestimmte Fähigkeiten zu erwerben und anzuwenden. Vorgesetzte müssen besonderen Wert darauf legen, dass die Mitarbeiterinnen selbständig entscheiden können, ob und in welchem Umfang sie Neues ausprobieren wollen. Die individuelle Entscheidung kann jedoch durch entsprechende Zielsetzungen gefördert werden. Zudem muss innerhalb der Organisation dafür gesorgt werden, dass die Mitarbeiter gut miteinander vernetzt sind – und zwar über die jeweiligen Einheiten hinweg und auch nach aussen – um ihnen die Möglichkeit zu geben, sich über die gemachten Erfahrungen auszutauschen. Hierzu muss eine Kultur vorherrschen, die offen mit Fehlern umgeht. Nur wenn aktives Fehlermanagement betrieben wird, können die Mitarbeiterinnen lernen, welche Methoden, Tools oder Vorgehensweisen funktionieren und welche nicht zum Erfolg führen. Hierzu dürfen Misserfolge im Rahmen der Evaluation neuer Möglichkeiten nicht negativ beurteilt werden. All dies gilt auch im Zusammenhang mit der Einführung eines benutzerzentrierten Entwicklungsprozesses. Da Innovationsfreudigkeit nicht themenspezifisch, sondern eine generelle Eigenschaft einer Organisation ist, wird diese Forderung in dieser allgemeinen Weise formuliert.

→ **Ein Problem- und ein Lösungsbewusstsein bezüglich Usability muss vorhanden sein.**

In der Anfangsphase, bevor Benutzerzentrierung als neues Entwicklungsparadigma eingeführt werden kann, muss zunächst das Bewusstsein vorhanden sein, dass Usability überhaupt ein wichtiges Qualitätsmerkmal eines Systems darstellt und dieses Qualitätsmerkmal oftmals nicht in ausreichendem Masse erfüllt wird (Phase I des Innovationsprozesses: [Wissensphase](#))

D.h. sowohl das Management als auch die Mitarbeiterinnen müssen zunächst ein Problembewusstsein entwickeln, dass die Usability des Endproduktes ein Erfolgskriterium von Software-Entwicklungsprojekten ist, das bisher nicht befriedigend erfüllt wird. Gleichzeitig müssen alle Mitglieder der Organisation, also sowohl das Management als auch alle im Entwicklungsprozess beteiligten Personen, überzeugt sein, dass Usability eine Systemeigenschaft ist, die systematisch entwickelt werden kann und die, genauso wie andere Qualitätsmerkmale, nicht dem Zufall überlassen werden darf.

Viele Organisationen beginnen ihre Usability Engineering Aktivitäten mit der Einrichtung eines Usability Labors, um mit Hilfe von Tests zu demonstrieren, dass die Usability der zu entwickelnden Produkte unzureichend ist. Dies ist eine zweischneidige Möglichkeit, um die Wissensphase zu unterstützen, da sie zu der weitverbreiteten Meinung führt, dass das Problem damit bereits gelöst sei (vgl. hierzu z.B. die Erfahrungen der Microsoft Corporation Usability Group (Dieli et al. 1994, S. 356). Eine andere Möglichkeit, die Wissensphase zu unterstützen, besteht z.B. in der Durchführung von Kosten-Nutzen Analysen von Usability Engineering Aktivitäten. Neben dem üblicherweise dargestellten ökonomischen Vorteil des Einsatzes von Usability Ressourcen (Karat 1994) (→ Aufzeigen einer Lösungsmöglichkeit für das Problem), zeigen Kosten-Nutzen Rechnungen nämlich bei entsprechender Kommunikation der Ergebnisse auch die Problemseite bei fehlender Benutzerzentrierung. Die Ergebnisse einer Kosten-Nutzen Analyse unterstützen damit gleichzeitig auch noch die Phase II des Innovationsprozesses: [Überzeugungsphase](#).

Eine weitere Möglichkeit zur Bewusstseinsbildung stellt eine Verbreitung von Normen zum Themenkreis Usability dar. Die Norm ISO 13407 beispielsweise verdeutlicht, dass es sich bei Benutzerzentriertheit nicht um eine exotische Idee handelt, sondern um eine weltweit anerkannte Anforderung an einen Entwicklungsprozess.

Ob ein entsprechendes Problem- und auch Lösungsbewusstsein innerhalb der Organisation und ihrer Mitglieder vorhanden ist, lässt sich z.B. daran abschätzen, ob und wie fehlende Usability innerhalb der Organisation als Mangel oder Risiko von Entwicklungsprojekten kommuniziert wird, ob es Bestrebungen gibt, um die Usability von Produkten gezielt zu fördern, ob Benutzerzufriedenheit ein angestrebtes Ziel von Entwicklungsprojekten ist und welche Erfolgsfaktoren für Entwicklungsprojekte formuliert werden.

→ **Grundlegendes Know-how bezüglich Benutzerzentrierung muss innerhalb der gesamten Organisation und speziell bei den zukünftigen Anwenderinnen der benutzerzentrierten Prozesse gefördert und verbreitet werden.**

Die Mitarbeiter müssen sich bewusst sein, dass Usability zu einem grossen Teil durch die Anforderungen der Endbenutzerinnen definiert wird und diese daher aktiv in den Entwicklungsprozess einbezogen werden müssen. Es muss ein gewisses Wissen auf allen Ebenen der Organisation und bei allen Mitarbeiterinnen vorhanden sein bezüglich der grundlegenden Eigenschaften benutzerzentrierter Entwicklungsprozesse.

Mitarbeiterinnen, die in Software-Entwicklungsprojekte involviert sind, müssen neben einem Problem- und Lösungsbewusstsein auch in die Lage versetzt werden, die Vorgaben benutzerzentrierter Prozesse zu evaluieren und sich selbständig von deren Sinn zu überzeugen (*evaluation knowledge*). Nur so können sie sich individuell für ein benutzerzentriertes Vorgehen entscheiden. Ausserdem müssen sie die Prozeduren und Konzepte von Benutzerzentrierung verstehen können (vgl. Phasen III und IV des Innovationsprozesses: [Entscheidungsphase](#) und [Implementationsphase](#)). Dies ist die Grundlage für die spätere effektive Umsetzung von benutzerzentrierten Prozessen (siehe hierzu auch Kap. 5.1.6 zum notwendigen Schulungswesen und Wissensmanagement einer Organisation).

Um das erforderliche *Grundlagenwissen* über benutzerzentrierte Entwicklungsprozesse in einer Organisation zu verbreiten, braucht es verschiedene Kommunikationskanäle, auf denen die Information bedarfsgerecht übermittelt wird. Es kann sich dabei z.B. um Informationen auf dem Intranet handeln, zu deren Lesen die Mitarbeiter explizit von ihrer Vorgesetzten aufgefordert werden, oder aber um Schulungen, Informationsveranstaltungen, Broschüren usw. Dieses Wissen, das der gesamten Organisation zur Verfügung stehen muss, ist entscheidend dafür, ob eine positive Attitüde gegenüber einem benutzerzentrierten Entwicklungsprozess entwickelt wird oder ob benutzerzentrierte Aktivitäten als nebensächlich und unnötiger Aufwand abgetan werden.

Detaillierteres Wissen kann über Durchführung von weithin sichtbaren Pilotprojekten erworben und kommuniziert werden. Da in diesem Fall die lokalen Kommunikationspfade von besonderer Bedeutung sind, sollten Pilotprojekte auch von Personen durchgeführt werden, die innerhalb der Entwicklungsorganisation gut vernetzt sind und die ein hohes Potential haben, ihre Erfahrungen weitreichend zu verbreiten.

→ **Das oberste Management muss sich klar zu einem benutzerzentrierten Entwicklungsprozess bekennen.**

Die Entscheidung einer Organisation zur Durchführung von benutzerzentrierten Entwicklungsprozessen muss mit einer eindeutigen Stellungnahme des obersten Managements einhergehen, um die bestmöglichen Voraussetzungen für eine erfolgreiche Einführung zu schaffen. Auflagen für Innovationen, dass diese (auch kurzfristig) keine zusätzlichen Kosten verursachen dürfen und sich innerhalb der Organisation und ihrer Abläufe nichts ändern darf, bedeuten, dass keine Innovation stattfinden kann. Um allen Mitgliedern einer Organisation die Bedeutsamkeit zu kommunizieren, muss das Ziel der Benutzerorientierung als strategisch wichtig definiert und gleichwertig mit anderen Zielen der Entwicklungsorganisation präsentiert werden. Nach Möglichkeit sollte die Rolle des «Usability-Champions» (siehe hierzu auch [Innovationschampion](#)) besetzt werden durch eine Person mit hohem Einfluss innerhalb der Entwicklungsorganisation, die sich sichtbar und in hohem Masse für die neuen Ideen einsetzt.

Eine Möglichkeit, diesem Commitment Ausdruck zu verschaffen sind z.B. vom obersten Management initiierte Programme, um die Benutzerzentrierung in der Organisation aktiv zu fördern. Diese Programme können aus der Entwicklung von Ausbildungsangeboten, Informationsveranstaltungen, Tools und Methoden oder auch aus Wettbewerben oder Anreizsystemen bestehen.

Die Glaubwürdigkeit der Entscheidung lässt sich z.B. daran messen, ob in einer Organisation die Qualifikation und Expertise in benutzerzentrierten Entwicklungsaktivitäten als genauso wertvoll erachtet werden, wie anderes Entwicklungs-Know-how. Ausserdem sind die für Usability Engineering Aktivitäten aufgewendeten Mittel im Vergleich zu Mitteln für die Erreichung anderer Qualitätsziele ein Indikator für die Gewichtung des Ziels «Benutzerzentriertheit».

→ **Benutzerzentriertheit muss allgegenwärtig sein.**

Einerseits zur Förderung der Glaubwürdigkeit, nicht zuletzt jedoch, um das Paradigma «Benutzerzentriertheit» wirklich konsequent und nachhaltig innerhalb der Organisation zu verankern, müssen auch die internen Abläufe einer Organisation ein gewisses Mass an Usability aufweisen. Das bedeutet, dass z.B. intern Informationen auf eine problem- und aufgabenorientierte Art und Weise angeboten werden oder dass die Entwicklungstools gebrauchstauglich sind.

Ein Mass für die Allgegenwärtigkeit von Benutzerzentriertheit einer Entwicklungsorganisation könnte sein, ob Usability ein Kriterium bei der Anschaffung von Standardsoftware ist.

Ein anderer Aspekt der Allgegenwärtigkeit von Benutzerzentriertheit ist die Einbeziehung von UE-Experten in Prozesse, die nicht unmittelbar mit der Entwicklung von Software zu tun haben. So ist z.B. die Entwicklung eines Corporate Designs (CD) eine Aufgabe, bei der Usability Fachleute in jedem Fall mit einbezogen werden sollten, nicht zuletzt deshalb, weil die Umsetzung eines CDs auf Benutzerschnittstellen von Softwareprodukten einen grossen Einfluss auf deren Usability hat.

→ **Die Forderung nach gebrauchstauglichen Produkten muss unterstützt werden.**

Nach wie vor suchen viele Benutzerinnen die Schuld eher bei sich selbst, wenn sie mit einem Softwareprodukt nicht zurecht kommen, anstatt ein gebrauchstauglicheres Produkt zu fordern. So lange die Situation so ist, dass die Bemühungen um benutzerfreundliche Produkte nicht tatsächlich verlangt werden, stehen Usability Engineering Aktivitäten auf verlorenem Posten:

„In many companies, usability has been the concern of a few human factors specialists, or has been taken up by a ‚usability champion‘. This has led to methods for improving usability to be supply-pushed by ‚experts, rather than demand-pulled by developers and end users. Until this balance shifts, usability in organizations will change but slowly however favourable other organizational factors may be. If there is no interest amongst users, developers and management, all usability speak will fall on deaf ears. So, no matter how much company support there is for usability, until it is demand-pulled by the other stakeholders in the organization, it will be performed half-heartedly and the benefits will not emerge.“ (Kaderbhai 1998, S. 36)

Dies bedeutet, dass Endbenutzerinnen darin unterstützt werden müssen, die Mangelhaftigkeit der Produkte als solche zu erkennen und entsprechend anzuprangern. Dies könnte dadurch erreicht werden, dass die Ergebnisse von Usability-Tests von Alltags-Software-Produkten (z.B. Tools zur Arbeitszeiterfassung oder Auftragsabwicklung) im Unternehmen im Intranet publiziert und diskutiert werden. So werden die Benutzer davon überzeugt, dass nicht sie ein Problem haben, sondern allenfalls die Produktqualität nicht ausreichend ist. Es bedeutet auch, dass die Klagen von Mitarbeitern über unbrauchbare Softwareprodukte ernst genommen werden und Massnahmen ergriffen werden, um gemeinsam mit den Endbenutzerinnen nach Alternativen oder Verbesserungsmöglichkeiten zu suchen.

→ **Die Erfolgsbeurteilung von Software-Entwicklungsprozessen muss Usability Aspekte und zumindest Teile der Nutzungsphase beinhalten.**

Die erfolgreiche Durchführung eines Entwicklungsprojektes verschafft den Projektleiterinnen und Projektbeteiligten in aller Regel Anerkennung innerhalb der Organisation. Diese kann sich z.B. in Form von Aufstiegsmöglichkeiten und/oder (finanziellen) Boni äussern. Die Art und Weise wie der Erfolg von Software-Entwicklungsprojekten gemessen wird, ist entscheidend dafür, ob benutzerzentrierte Aktivitäten als Chance oder als Hemmschuh wahrgenommen werden. Wird als Erfolgskriterium ausschliesslich die Einhaltung der vorgegebenen Zeit und des vorgegebenen Budgets bis zum lauffähigen Produkt gewertet, wird die Einführung von benutzerzentrierten Entwicklungsprozessen schwierig bleiben, da Usability Engineering Aktivitäten dieser Definition von Erfolg scheinbar zuwiderlaufen. Erst wenn die Usability des Endproduktes ein gleichwertiges Erfolgskriterium ist wie die Einhaltung der Zeit- und Mittelvorgaben, sind die zusätzlichen Aufwendungen, die ein benutzerzentrierter Prozess möglicherweise erfordert, in der Projektlogik gerechtfertigt.¹² In Organisationen, in denen In-house Software entwickelt wird, scheint der Kostendruck nicht ganz so gross zu sein, wie für Entwicklungen für den kompetitiven Markt. Weder die empfundenen Zusatzkosten noch die vermuteten finanziellen Vorteile von Usability Aktivitäten, die ohnehin schwierig zu quantifizieren sind, mögen daher in diesem Fall sehr bedeutend sein. Vielmehr scheint die Priorisierung der Ziele: grosse Benutzerzahlen und hohe Benutzerinnenzufriedenheit innerhalb der Entwicklungsorganisation für die erfolgreiche Einführung von Usability Aktivitäten relevant (Kaderbhai 1998).

Um einen Entwicklungsprozess zu implementieren, der die Benutzerinteressen als zentrale Anforderungen berücksichtigt, muss auch die Erfolgskontrolle entsprechend ausgeweitet werden. Im Fall des Qualitätsmerkmals «Usability» besteht jedoch die Schwierigkeit, dass sich der Erfolg nicht unmittelbar mit der Fertigstellung des Produktes, sondern erst im Laufe einer bestimmten Nutzungsphase ermitteln lässt. Daher muss der Begriff des Erfolgs eines Software-Entwicklungsprojektes sowohl inhaltlich als auch zeitlich ausgedehnt werden, wobei die Erfolgsbeurteilung aus offensichtlichen Gründen nicht bis zum Ende des

¹² Die Einschätzung, dass Usability Engineering Aktivitäten einen zusätzlichen Aufwand bedeuten, ist mit hoher Wahrscheinlichkeit nicht gerechtfertigt. Im Hinblick auf den gesamten Lifecycle eines Produktes hilft die Verbesserung der Usability, die Verkaufszahlen zu erhöhen oder bei interner Entwicklung, z.B. die Aufwendungen für Schulungen zu vermindern. Geht man davon aus, dass ca. 80% der entstehenden Kosten innerhalb des Software-Lifecycles in der Pflegephase nach dem ersten Release auftauchen und davon wiederum ca. 80% aufgrund von unvorhergesehenen Benutzeranforderungen entstehen (Karat 1994), kommt man zu dem Schluss, dass hier das grösste Einsparungspotential besteht und dass dieses die anfallenden Kosten vermutlich bei weitem übersteigt. Der potentielle Nutzen von Usability wird z.B. ausführlich bei (Donahue et al. 1999) beschrieben.

Lifecycles des Produktes warten kann. Eine pragmatische Herangehensweise wäre, die Erfüllung der Usability Ziele mittels eines formalen Usability Tests zum Abschluss der Entwicklung zu prüfen und der eigentlichen Ausbreitung einer Applikation eine Pilotphase vorausgehen zu lassen, in deren Rahmen dann die Benutzerzufriedenheit und der Nutzungsumfang ermittelt werden kann.

→ **Anreizsystem müssen die Verbreitung von Benutzerzentriertheit belohnen.**

Aufgrund der Innovationsforschung weiss man, dass Anreizsysteme, die dazu dienen, Personen zur Annahme einer bestimmten Innovation zu überreden, ohne dies aus eigener Überzeugung zu tun, problematisch sind. Gerade im Falle benutzerzentrierter Entwicklungsaktivitäten ist es von grosser Bedeutung, dass die involvierten Personen vom Sinn und Zweck der Aktivitäten überzeugt sind, wenn diese den gewünschten Erfolg erzielen sollen. So lässt sich beobachten, dass z.B. die Ergebnisse eines Usability Tests, wenn dieser ‚von oben‘ aufgezwungen wird, nur sehr unwillig und begrenzt umgesetzt werden. Mangelt es den Anwendern einer Innovation an eigener Überzeugung, leidet häufig die Effektivität. Aus diesen Überlegungen lässt sich ableiten, dass Anreizsysteme allenfalls so gestaltet sein müssen, dass sie die Personen belohnen, die Dritte von der Durchführung benutzerzentrierter Aktivitäten (auf freiwilliger Basis) überzeugen.

→ **Fehlende Benutzerzentrierung im Entwicklungsprozess wird sanktioniert.**

Im Fall eines derart erfolgskritischen Qualitätsmerkmals des Endproduktes, wie es die Usability darstellt, könnte auch der Einsatz eines Anreizsystems gerechtfertigt sein, das nicht belohnt, sondern die Vernachlässigung der Benutzerinteressen sanktioniert. Auch eine Kombination aus Belohnung und Sanktion ist denkbar.

→ **Es müssen abteilungsübergreifende Ziele und eine gemeinsame Erfolgsbeurteilung vorhanden sein.**

Innerhalb der Software-Entwicklerzunft besteht eine wachsende Diversität. Die einen sind spezialisiert auf die Erhebung von technischen Anforderungen, die anderen auf das Architekturdesign, Dritte auf die Ausprogrammierung von Code und Komponenten usw. Doch Software-Entwicklung ist längst keine rein technische Aufgabe mehr. Zunehmend werden auch die Auftraggeber in die Pflicht genommen, die beispielsweise gemeinsam mit dem Entwicklungsteam funktionale und leistungsbezogene Anforderungen spezifizieren müssen, die dann die Grundlage für einen Vertrag werden. Marketingleute oder Businessexpertinnen werden einbezogen um zu definieren, welche Funktionalitäten gebraucht werden, Prozessingenieure erarbeiten neue Arbeitsabläufe, Designer werden hinzugezogen, um die Benutzerschnittstelle ansprechend zu gestalten und Usabilityexpertinnen entwickeln konzeptuelle Modelle für die Benutzerschnittstelle. Das Zusammentreffen derart verschiedener Disziplinen ist eine Herausforderung. Da die Beteiligten nicht zwangsläufig Mitglieder eines einzigen Teams sind, fallen Schnittstellen an, über die hinweg kommuniziert werden muss. Dies wird dadurch erschwert, dass die verschiedenen Projekt-«Stakeholder» unterschiedliche Sprachen sprechen und z.T. konkurrierende Interessen verfolgen. Um den Weg für diese schwierige Aufgabe zu ebnen, müssen konkrete Massnahmen ergriffen werden, die das Zusammengehörigkeitsgefühl stärken, z.B. in Form von gemeinsam formulierten Zielen, für deren Erreichung alle Beteiligten verantwortlich gemacht werden. Auf jeden Fall muss verhindert werden, dass der Erfolg einer Software (z.B. in Form hoher Verkaufszahlen) nur einer Gruppe (z.B. dem Marketing) angerechnet wird, während die Aufwendungen für eine konsequente Benutzereinbeziehung der IT-Abteilung belastet werden.

5.1.5 Qualifizierte Usability Engineering Mitarbeiter und Mitarbeiterinnen

Die hauptsächliche Funktion der Usability Engineering (UE) Einheit in einer Organisation ist es, das entsprechende Know-how bereitzustellen und zu vermehren, das zur Durchführung aller Aktivitäten im Zusammenhang mit der Einbeziehung von Benutzerinnen in den Entwicklungsprozess benötigt wird.

→ **Die UE-Einheit der Organisation muss in der Lage sein, alle benutzerzentrierten Aktivitäten des Entwicklungsprozesses fachgerecht durchzuführen.**

Auch wenn in jedem Projekt Personen festgelegt werden müssen, die für den Usabilityaspekt verantwortlich sind, kann man nicht davon ausgehen, dass diese auch automatisch Usability Experten sind, die selbständig alle anfallenden Aufgaben erfüllen können. Die UE-Einheit muss jedoch in der Lage sein, die Projekte in allen Phasen zu unterstützen und entweder bei der Durchführung der Aktivitäten zu coachen oder diese bei Bedarf selbst durchführen. Ausserdem muss die UE-Einheit befähigt sein, für jedes Projekt diejenigen Methoden und Vorgehensweisen zu definieren, die im spezifischen Kontext am adäquatesten sind.

→ **Die mit benutzerzentrierten Aktivitäten betrauten Personen müssen hierzu qualifiziert sein und diese Qualifikation nachweisen können.**

Möglichst alle Projektbeteiligten sollten die Gelegenheit haben, Kontakt mit realen Benutzern zu bekommen. Doch nicht alle Personen sollten auch tatsächlich in grösserem Umfang benutzerzentrierte Aktivitäten durchführen, ohne hierfür wirklich qualifiziert zu sein. Die unsachgemässe Einbeziehung von Benutzerinnen in den Entwicklungsprozess ist nicht nur wenig hilfreich, sie kann auch dazu beitragen, dass die Entwicklung erschwert wird und die Benutzer aufgrund eines frustrierenden Erlebnisses die Zusammenarbeit für spätere Projekte ablehnen. So ist es z.B. nicht erstrebenswert, dass die einbezogenen Benutzer über Designfragen entscheiden oder das Gefühl erhalten, dass sie selbst definieren müssten, wie die Abläufe in der neuen Applikation sein sollen. Dies kann und soll nicht der Fall sein, da Benutzerinnen zum einen schlecht in der Lage sind, genau zu äussern, was sie für eine optimale Unterstützung ihrer Tätigkeit benötigen und vor allem jeder Benutzer andere Ideen und Vorstellungen von ‚seinem‘ Tool haben wird. Um zu garantieren, dass alle benutzerzentrierten Aktivitäten systematisch zu einer Optimierung der Benutzerschnittstelle beitragen, müssen die Durchführenden entsprechende Qualifikationen z.B. im Bereich des Usability Engineerings, der Ethnologie oder der Psychologie aufweisen. Diese sollen sowohl theoretische Kenntnisse als auch praktische Erfahrung mit dem Thema umfassen. Der Usability Verantwortliche innerhalb eines Projektes muss den Usability Engineering Prozess und seine zugrundeliegenden Prinzipien sowie den Sinn jeder einzelnen Aktivität kennen und in der Lage sein, die anfallenden Aufgaben zu koordinieren. Dieses Wissen kann er sich durch den Besuch von internen oder externen Schulungsmöglichkeiten erwerben (siehe hierzu auch den Abschnitt 5.1.6 zum Schulungswesen und Wissensmanagement).

→ **Alle Personen, die benutzerzentrierte Aktivitäten ausüben oder koordinieren, müssen sich kontinuierlich weiterbilden.**

Mit dem einmaligen Erwerb von Sachkenntnis im Bereich der benutzerzentrierten Software-Entwicklung ist es nicht getan. Durch die Anwendung von benutzerzentrierten Methoden innerhalb eines bestimmten Entwicklungskontextes werden zunehmend Erfahrungen gesammelt und Vorgehensweisen optimiert. Sowohl der Prozess als auch die zur Verfügung stehenden Methoden sind einem kontinuierlichen Wandel unterworfen und es muss die Möglichkeit bestehen, aus den Erfahrungen auch zu lernen. Die UE-Expertinnen müssen intensiven Erfahrungsaustausch pflegen und dieses Wissen auch innerhalb der Organisation kommunizieren.

→ **Die zentrale UE-Einheit muss innerhalb der Fachwelt vernetzt sein.**

Das Gebiet des Usability Engineerings in der Praxis ist noch relativ neu. Obwohl schon lange zu diesem Thema publiziert wird, gibt es zumindest im deutschsprachigen Raum noch wenige Organisationen, in denen tatsächlich Usability Engineering Prozesse etabliert werden konnten, die über einen Prototypstatus hinausgehen. In solch einem Umfeld ist der Austausch von Erfahrungen über die Grenzen der eigenen Organisation hinweg besonders wichtig. Daher sollten zumindest die Mitglieder einer UE-Einheit in eine übergeordnete Struktur, wie beispielsweise eine nationale SIGCHI (Special Interest Group Computer Human Interaction) eingebunden sein oder den Kontakt zu einer Universität suchen, um eine entsprechende Diskussion auf übergeordneter Ebene führen zu können.

5.1.6 Ein leistungsfähiges Schulungswesen und Wissensmanagement

Eine Organisation, die erfolgreich benutzerzentriert Software entwickeln will, muss den Projekten zur Durchführung der entsprechenden Aktivitäten auch das notwendige Know-how zur Verfügung stellen. Die einzelnen Teammitglieder sollen die Möglichkeit haben, auf Wissensquellen innerhalb der Organisation zuzugreifen, die es ihnen erlauben, bedarfsgerecht Informationen abzufragen. Diese Informationen müssen sowohl Grundlagen- als auch Praxiswissen beinhalten, das für eine direkte Anwendung innerhalb eines Entwicklungs-Prozesses brauchbar ist.

→ **Das Schulungswesen muss sowohl das organisationelle als auch das individuelle Lernen bezüglich Benutzerzentrierung unterstützen.**

Das in der Organisation vorhandene Wissen sollte im Sinne eines organisationellen Lernprozesses systematisch erweitert werden. Hierzu sollten die Erfahrungen, die innerhalb oder ausserhalb der Organisation mit der Anwendung von benutzerzentrierten Vorgehensweisen gemacht werden, in Wissensdatenbanken gesammelt und ausgetauscht werden. Das Methodenangebot sollte von zentraler Stelle (UE-Team) bedarfsgerecht (und angepasst an den vorgeschriebenen Entwicklungsprozess) erweitert und gepflegt werden. Alle für die Durchführung von benutzerzentrierten Aktivitäten Verantwortlichen sollten Zugang zu Schulungen haben. Zudem sollte die Möglichkeit bestehen, neue Methoden zunächst unter Aufsicht eines erfahrenen Usability Engineers und Coaches durchzuführen, um ‚Learning on the Job‘ zu ermöglichen. Nach

Möglichkeit sollte zum Abschluss von Projekten ein Austausch über gute und schlechte Erfahrungen mit der Anwendung von benutzerzentrierten Methoden stattfinden, der von einer Usability Expertin moderiert wird.

→ **Das verfügbare Wissen über benutzerzentrierte Aktivitäten sollte in einer praxisnahen Form angeboten werden.**

Die Anwendung von Usability Know-how ist komplex. Auf Designebene gibt es keine perfekten Lösungen und jede Entscheidung besteht aus dem Abwägen von Vor- und Nachteilen. Dies ist schwer in einer abstrakten Form zu kommunizieren. Es gibt inzwischen verschiedenste Ansätze, um das verfügbare theoretische und praktische Usability Wissen anzuwenden und auch wiederverwenden zu können. Der häufigste Ansatz ist vermutlich die Arbeit mit Guidelines und Styleguides, über die das eigentliche User Interface Design erleichtert werden soll. Diese Vorgehensweise hat jedoch den Nachteil, dass Lösungsvorschläge nicht in einen konkreten Projektkontext gestellt werden können und es dadurch unmöglich ist zu entscheiden, ob die spezifische Problemstellung tatsächlich durch ein bestimmtes Design gelöst werden kann. Erschwerend kommt hinzu, dass Guidelines oftmals zu abstrakt, zu vereinfachend oder zu schwierig zu interpretieren sind. Ein Lösungsvorschlag, um die oben beschriebenen Probleme zu umgehen, ist die Arbeit mit sogenannten Usability Patterns. Durch die Beschreibung des jeweiligen Anwendungskontextes für ein spezifisches Design (oder durch die Beschreibung eines Kontextes, für den das Design nicht erfolgsversprechend ist → Anti-Pattern) wird die gezielte Anwendung von Designvorschlägen ermöglicht (van Welie et al. 1999). Eine Erweiterung von Patterns sind sogenannte Claims, die Designerinnen oder Entwickler von Benutzerschnittstellen dazu anhalten sollen, die Vor- und Nachteile verschiedener Lösungen abzuwägen. Ausführliche Erläuterungen zu diesem Thema finden sich in Kap. 7.2.1.

Wird der Aufgabenrahmen des Usability Engineerings noch weiter aufgespannt, muss nicht nur das Interface Design, sondern auch alle hierzu notwendigen Vor- und Nacharbeiten methodisch unterstützt werden. Auch hierzu gibt es bereits Ideen, wie auf der Basis eines gegebenen Projektkontextes mit Hilfe eines Tools die Auswahl geeigneter Methoden unterstützt werden kann (Metzker et al. 2001). In erster Linie geht es bei all diesen Ansätzen darum, die Komplexität des Themas Usability Engineering dadurch zu reduzieren, dass mögliche Lösungen aufgrund der Beschreibung eines Anwendungskontextes eingegrenzt werden und so die Anwendung auch für weniger erfahrene Mitarbeiterinnen erleichtert wird. Um erfolgreich Usability Know-how in einer Entwicklungsorganisation verbreiten zu können, muss das vorhandene Wissen auf Methoden-, Prozess- und Designebene kontextbezogen präsentiert werden.

→ **Der Einsatz von Methoden und Tools zur benutzerzentrierten Software-Entwicklung muss über einen Reviewprozess validiert werden.**

Nur wenn die Erfahrungen, die mit gewissen Vorgehensweisen in bestimmten Kontexten gemacht werden, rückgespeist und ausgewertet werden, kann auf der Ebene der Organisation ein Lernprozess stattfinden. Daher müssen Prozesse implementiert werden, die diesen Ringschluss aus Lernen, Dokumentieren, Abfragen und Kommentieren fördern.

→ **Die systematische Erfassung, Verwaltung und Verfügbarmachung von usabilityrelevantem Know-how sollte durch ein Tool unterstützt werden.**

Wissensmanagement ist in den letzten Jahren ein heisses Thema geworden. In grossen, oftmals räumlich und zeitlich verteilt arbeitenden Organisationen ist es nahezu unmöglich, den Überblick über das vorhandene Wissen und die vorhandenen Erfahrungen zu behalten. Will man in einer solchen Organisation das Wissen im Hinblick auf benutzerzentrierte Entwicklungsprozesse optimal sammeln und verfügbar machen, ist hierzu die Verwendung eines Tools sinnvoll. In jüngster Zeit sind in diesem Bereich einige interessante Ansätze publiziert worden (Henninger 2001b), (Metzker 2001), die in Kap. 7.2 diskutiert werden.

5.1.7 Ausreichende Ressourcen für die übergeordneten Aufgaben

Die Frage nach ausreichenden Ressourcen ist indirekt bereits in fast allen anderen Dimensionen des Reifegradmodells beinhaltet. Es ist jedoch sinnvoll, Ressourcen als eigenständige Dimension bei der Beurteilung der Benutzerzentrierung von Organisationen heranzuziehen, da so die Möglichkeit gegeben wird, diese Frage pointiert zu thematisieren.

→ Ausreichende Ressourcen zur Pflege des Entwicklungsprozesses.

Obwohl jedes einzelne Projekt einen massgeschneiderten Prozess benötigt, muss auch dafür gesorgt werden, dass das generische Prozessmodell mit den Erfahrungen, die innerhalb der Organisation gemacht werden, mitwächst und optimiert wird. Dadurch fallen auf der Ebene des Prozesses kontinuierlich Aufgaben zur Entwicklung und Pflege der Usability Engineering Aktivitäten und deren Integration in den vorhandenen Entwicklungsprozess an. Diese sollten zwar nicht kontinuierlich umgesetzt werden, aber Veränderungswünsche oder Optimierungsbedarf bzw. –möglichkeiten sollten permanent gesammelt werden.

→ Ausreichende Ressourcen zur Aufrechterhaltung der methodischen und physischen UE-Infrastruktur und des Know-hows.

Die UE-Einheit ist für zahlreiche Aufgaben zuständig, die nicht unmittelbar zur Produktivität beitragen. Z.B. sollten hier zentral Guidelines und Styleguides für die gesamte Entwicklungsorganisation erstellt werden (Normierungsfunktion), das vorhandene Know-how muss hier gesammelt und verfügbar gemacht werden, die Methoden müssen entwickelt (oder zumindest angepasst) und ausprobiert werden usw. Dies ist nur möglich, wenn die Mitglieder der zentralen UE-Einheit hierfür einen substanziellen Teil ihrer zeitlichen Ressourcen verwenden können und zudem Mittel zur Verfügung haben, um Konferenzen und Weiterbildungen zu besuchen, Fachtagungen zu organisieren usw.

Ausserdem muss für die physische Infrastruktur gesorgt sein. Ob eine UE-Einheit nun mehrere Usabilitylabore, Designräume und Schulungsräume benötigt, sei dahingestellt. Wird jedoch in einer grossen Entwicklungsorganisation institutionalisiert benutzerzentriert Software entwickelt, ist es sinnvoll, früher oder später ein Usabilitylabor einzurichten. Als Mindestanforderung ist zu nennen, dass die UE-Einheit die Möglichkeit hat, sich an einem repräsentativen Ort, möglichst in räumlicher Nähe zu Entwicklungseinheiten, anzusiedeln. Es sollte zumindest soviel Raum vorhanden sein, dass Teamarbeiten und kreative Sitzungen möglich sind.

→ Ausreichende Ressourcen für die interne Ausbildung.

Die Usability Engineering Einheit sollte ggf. in Zusammenarbeit mit internen oder externen Schulungskräften Schulungsmodule ausarbeiten, um die Idee der Benutzerzentrierung immer wieder bei den Mitarbeiterinnen zu verbreiten, neue Mitarbeiter einzuführen, Entwicklerinnen in neuen Methoden zu schulen usw. Hierzu müssen die entsprechenden Ressourcen zur Verfügung stehen.

→ Ausreichende Ressourcen für Gremientätigkeiten.

Aktivitäten ausserhalb der eigentlichen Software-Entwicklungsabteilung haben z.T. Einfluss auf die Usability der zu entwickelnden Software. Wird z.B. von der Marketingabteilung ein Corporate Design (CD) erstellt, das in allen Softwareprodukten eingehalten werden soll, ist es wichtig, dass Usabilityaspekte bei der Schaffung eines CDs berücksichtigt werden. Ebenso kann es sinnvoll sein, dass Usability Expertinnen einbezogen sind in technische Entscheidungen mit Auswirkungen auf die GUI Entwicklung, in das Prozessdesign, in Managementaktivitäten usw., um Usability Belange innerhalb und ggf. auch ausserhalb der Organisation erfolgreich zu kommunizieren.

→ Ausreichende Ressourcen für die Unterstützung aller relevanten Projekte.

Das längerfristige Ziel aller Bemühungen in Richtung ‚benutzerzentrierte Organisation‘ ist es, nicht nur in Einzelfällen, sondern organisationsweit benutzerzentriert Software zu entwickeln. Dies bedeutet jedoch auch, dass die Usability Einheit in der Lage sein muss, die steigende Anzahl Projekte zu bedienen, da trotz Schulungen und Hilfsmittel nicht davon ausgegangen werden kann, dass die Projekte alle benutzerzentrierten Aktivitäten vollkommen selbständig ausführen können. Hierzu muss entweder die Usability Engineering Einheit oder aber die Anzahl der fest in der Entwicklungsabteilung ansässigen Mitarbeiterinnen mit Usability Hintergrund zusammen mit dem Bedarf wachsen. Diese Anforderung ist umso wichtiger, da es sehr kontraproduktiv ist, wenn erstmalig anfragende Projekte nicht bedient werden können oder benutzerzentrierte Aktivitäten mangels kompetenter Unterstützung unsachgemäss durchgeführt werden und daher erfolglos bleiben.

→ Ausreichende Ressourcen für die Hinzuziehung externer Expertinnen.

Aus verschiedenen Gründen kann es sinnvoll sein, hin und wieder externe Usability Fachkräfte hinzuzuziehen. Sei es nun, um einen Wissenstransfer zu erzielen, um einen unabhängigen Dritten in strittigen Entscheidungen hinzuzuziehen oder um sich von jemandem mit ausreichender Distanz und ohne Betriebsblindheit in strukturellen Fragen beraten zu lassen. Externe Consultants – zumal die Guten – sind eine teure Angelegenheit und dennoch ist es erstrebenswert, bei Bedarf die Möglichkeit zu haben, auf externe Expertinnen zurückzugreifen.

5.2 Das Ratingverfahren

Die Bedürfnisse und Geschäftsziele einer Organisation beziehen sich in der Regel auf eine Verbesserung der Kundenzufriedenheit und eine höhere Wettbewerbsfähigkeit. Im Falle von Software-Entwicklungsorganisationen bedeutet dies, dass sich das Hauptaugenmerk des Managements auf eine Verbesserung des Entwicklungsprozesses richtet, um Ziele wie eine höhere Produktqualität, niedrigere Entwicklungs- und Pflegekosten, eine kürzere Entwicklungszeit und eine bessere Vorhersagbarkeit und Kontrollmöglichkeit bezüglich des Prozesses und des Produktes zu erhalten. Um einen Entwicklungsprozess kontinuierlich verbessern zu können, ist es notwendig, den Prozess zu bewerten und die Effektivität des Prozesses hinsichtlich der formulierten Ziele zu messen (siehe Beispiel CMM, Kap. 2.2.1).

Bei einem Gesamtkonzept zur Förderung eines benutzerzentrierten Software-Entwicklungsvorgehens geht es nicht nur um den Entwicklungsprozess. In erster Linie wird hierbei angestrebt, den Boden innerhalb der Organisation zu bereiten, damit ein benutzerzentrierter Entwicklungsprozess überhaupt eine Chance auf Erfolg hat. Die Faktoren, die hierbei eine Rolle spielen sind vielfältig und wurden im Kap. 4.1 ausgiebig dargelegt und begründet. Um die zahlreichen Anforderungen an eine ‚benutzerzentrierte Entwicklungsorganisation‘ zu erfüllen, braucht es mehr als einen Managemententscheid. Es wurde dargelegt, dass es sich bei der ‚Idee Benutzerzentrierung‘ um eine Art Paradigma handelt, das im Vergleich zu den konventionellen Entwicklungsvorgehen eine Innovation darstellt, deren Implementation nicht von heute auf Morgen möglich ist. Mayhew hat dies in Bezug auf ihren Usability Engineering Lifecycle folgendermassen formuliert: „(.) bear in mind that you will not likely be able to take a development organization with little or no prior experience with the Usability Engineering Lifecycle and convince them to try – and then succeed in – implementing the whole Usability Engineering Lifecycle (..) *overnight*. Instead, you will need to take small, strategic steps in leading the organization to evolve gradually towards integrating a fullblown Usability Engineering Lifecycle into their standard development methodology.“ (Mayhew 1999, S. 423)

Die Idee dieser Arbeit ist es, alle anderen Bemühungen – neben dem eigentlichen Entwicklungsprozess – gleichgewichtig anzugehen und sich nicht nur auf die eigentlichen Usability Engineering Methoden zu stützen. Dies soll helfen, die ‚Reaktionsträgheit‘ einer Organisation gegenüber benutzerzentrierten Ansätzen zu überwinden, um so messbare und nachhaltige Schritte hin zu einer benutzerzentrierten Entwicklungsorganisation machen zu können.

Hierzu wird ein Ratingverfahren eingesetzt, das es erlauben soll, abzuschätzen, wie weit eine Organisation im Hinblick auf die Benutzerzentrierung fortgeschritten ist. Die Evaluation wird für jede Dimension einzeln vorgenommen. Dadurch erhält beispielsweise ein Consultant Hinweise, in welchem Bereich weitere Anstrengungen sinnvoll sein könnten, um die Benutzerzentrierung zu fördern. Der Stand der Benutzerzentrierung einer Organisation wird in 3 Stufen eingeteilt (zuzüglich der Stufe 0 für kaum oder gar keine Benutzerzentrierung). Diese Einteilung ist angelehnt an den DATech-Prüfbaustein: Usability Engineering Prozess. Letztlich ist eine derartige Einteilung willkürlich und muss sich daran messen lassen, ob sie für Handlungsanleitungen geeignet ist. Diesbezüglich scheinen 3 Stufen – anstelle von 5-stufigen Ansätzen, wie beispielsweise CMM – ausreichend. Da es sich nicht um ein Vorgehen zur Zertifizierung einer Organisation handelt, ist eine Zusammenfassung der Einzelbewertungen der 7 Dimensionen in ein abschliessendes Gesamtergebnis nicht notwendig. Dies wäre ohnehin wenig sinnvoll, da damit die eigentliche Idee, nämlich einen systematischen Blick auf einzelne Dimensionen zu erlauben, ad absurdum geführt würde.

5.2.1 3 Reifestufen für jede Dimension

Im folgenden wird kurz skizziert, wodurch sich Organisationen auf den verschiedenen Reifestufen hin zu einer benutzerzentrierten Organisation auszeichnen:

Auf **Stufe 1** befindet sich eine Organisationen, in der auf Managementebene oder auf der Ebene einzelner, für die Software-Entwicklung verantwortlicher Personen, die Überzeugung vorhanden ist, dass die Durchführung von benutzerzentrierten Aktivitäten sinnvoll und für einen längerfristigen Erfolg der Entwicklungsprojekte notwendig ist. Die Überzeugung äussert sich darin, dass erste Schritte hin zur Benutzerzentrierung unternommen wurden und auch bereits erste Erfahrungen mit benutzerzentrierten Vorgehen gesammelt wurden, die dazu dienlich sind, die Idee innerhalb der Organisation weiter zu verbreiten. Es wurden erste Massnahmen zur Schaffung einer Usability Infrastruktur unternommen und Usability Experten ins Boot geholt, die in der Lage sind, diese Massnahmen in geeigneter Weise zu propagieren.

Auf **Stufe 2** ist Benutzerzentrierung nicht mehr nur ein informelles Projekt innerhalb der Organisation, sondern bereits in Form sichtbarer Strukturen, grösstenteils mit engem Bezug zur eigentlichen Software-Entwicklung, implementiert. Die Basis für die Durchführung kompletter benutzerzentrierter Software-Ent-

wicklungsprozesse ist vorhanden. Im Entwicklungsprozess der Organisation sind benutzerzentrierte Aktivitäten aufgeführt, die den gesamten Prozess abdecken und es gibt ein geeignetes Methodenportfolio und auch ausreichend qualifizierte Mitarbeiterinnen, um die benutzerzentrierten Aktivitäten sachgerecht durchführen zu können. Benutzerzentrierung ist ein wichtiges Thema innerhalb der Organisation geworden und alle Mitglieder der Organisation wissen, was sich hinter dem Begriff Usability verbirgt. Der Erfolg der entwickelten Produkte steht und fällt mit ihrer Usability und die Zahl der Projekte, in denen benutzerzentrierte Aktivitäten durchgeführt werden, steigt rasch an.

Auf **Stufe 3** ist Benutzerzentrierung organisationsweit umgesetzt. Der Entwicklungsprozess umfasst den gesamten Lifecycle von Produkten und benutzerzentrierte Aktivitäten sind vollständig und gleichwertig mit anderen qualitätssichernden Massnahmen integriert. Es ist eine Organisationsstruktur vorhanden, die es erlaubt, alle Projekte effizient und effektiv mit Usability Know-how zu versorgen. Das Wissen bezüglich benutzerzentrierter Vorgehensweisen ist hoch und diese werden selbstverständlich umgesetzt. Die Organisation ist in der Lage, ihre Erfahrungen zu einer systematischen Verbesserung der Vorgehensweisen und Prozesse einzusetzen und organisationsweit genießt die Usability von Abläufen und Produkten einen hohen Stellenwert.

Die im vorigen Kapitel formulierten Forderungen werden nun zu Reifemerkmale umformuliert, die zur Erreichung der 3 Stufen erfüllt sein müssen. Der Aufbau berücksichtigt auch Überlegungen zu einer sinnvollen Vorgehensweise bei der Umsetzung von Benutzerzentrierung.

Dimension	Anforderungen	Reifegrad
Prozessmodell	Es gibt Prozesskategorien, dazugehörige Aktivitäten und quantifizierbare Ziele bezüglich Benutzerzentrierung.	1
	Die Aktivitäten sind methodenneutral formuliert, verweisen jedoch auf geeignete Methoden.	1
	Das Prozessmodell beinhaltet die Evaluation der Usability des Endprodukts.	1
	Das Prozessmodell unterstützt aktiv Iteration.	2
	Das Prozessmodell vereinigt Usability Engineering und Software-Engineering Prozesse.	2
	Verantwortlichkeiten im Prozess werden über ein Rollenkonzept definiert, das die Benutzerzentrierung explizit abdeckt.	2
	Das Dokumentenhandling ist effizient und effektiv.	2
	Der Prozess deckt den kompletten Produkt-Lifecycle ab.	2
	Die Projektdokumentation ermöglicht die systematische Aufarbeitung von usabilityrelevantem Wissen.	3
	Das Prozessmodell ist skalierbar und modifizierbar.	3
Die Präsentation des Prozessmodells ist gebrauchstauglich.	3	

Tabelle 1: Anforderungen der 7 Dimensionen an jede Reifegradstufe

Dimension	Anforderungen	Reifegrad
Methoden- und Toolangebot	Das Methoden- und Toolangebot ist geeignet, um alle benutzerzentrierten Aktivitäten im Prozessmodell unterstützen zu können (Vollständigkeit).	1
	Die Methodenauswahl und die Methoden selbst sind an das jeweilige Projekt anpassbar.	2
	Die Entwicklungsmethoden und –Tools sind bewährt und an die speziellen Gegebenheiten der Organisation angepasst.	3
Organisationsstruktur	Die UE-Experten können sowohl innerhalb der Projekte als auch als Consultant und Coach tätig sein.	1
	Die UE-Einheit hat ein zentrales Standbein innerhalb der Organisation.	2
	Die UE-Einheit einer ‚In-house‘ entwickelnden Organisation ist so platziert, dass sie sowohl IT initiierte als auch Business initiierte Entwicklungsprojekte unterstützen kann.	2
	Ein multidisziplinäres Team aus Business-Verantwortlichem, UE-Expertin, Interface-Entwickler und Interface-Designerin hat die Entscheidungsbefugnis über die GUI-Entwicklung.	3
Managementaktivitäten und Kultur	Es ist bei allen Entwicklungsteams ein Problem- und ein Lösungsbewusstsein bezüglich Usability vorhanden.	1
	Das oberste Management bekennt sich klar zu einem benutzerzentrierten Entwicklungsprozess.	1
	Die Innovationsfreudigkeit der Mitarbeiter wird gefördert.	1
	Die Forderung nach gebrauchstauglichen Produkten wird unterstützt.	2
	Grundlegendes Know-how bezüglich Benutzerzentrierung wird innerhalb der gesamten Organisation und speziell bei den zukünftigen Anwenderinnen der benutzerzentrierten Prozesse gefördert und verbreitet.	2
	Die Erfolgsbeurteilung von Software-Entwicklungsprozessen beinhaltet Usability Aspekte und umfasst auch Teile der Nutzungsphase.	2
	Anreizsysteme belohnen das Verbreiten von Benutzerzentriertheit.	2
	Benutzerzentriertheit ist allgegenwärtig.	3

Tabelle 1: Anforderungen der 7 Dimensionen an jede Reifegradstufe - Fortsetzung

Dimension	Anforderungen	Reifegrad
	Es ist ein organisationsweites Problem- und ein Lösungsbewusstsein bezüglich Usability vorhanden.	3
	Fehlende Benutzerzentrierung im Entwicklungsprozess wird sanktioniert.	3
	Es sind gemeinsame abteilungsübergreifende Ziele und eine gemeinsame Erfolgsbeurteilung vorhanden.	3
Qualifikation der Mitarbeiter	Die UE-Einheit der Organisation ist in der Lage, alle benutzerzentrierten Aktivitäten des Entwicklungsprozesses fachgerecht durchzuführen.	1
	Die zentrale UE-Einheit ist innerhalb der Fachwelt vernetzt.	2
	Alle Personen, die benutzerzentrierte Aktivitäten ausüben oder koordinieren, bilden sich kontinuierlich auf diesem Gebiet weiter.	2
	Alle innerhalb der Organisation mit benutzerzentrierten Aktivitäten betrauten Personen sind hierzu qualifiziert und können diese Qualifikation nachweisen.	3
Schulungswesen und Wissensmanagement	Das Schulungswesen unterstützt das individuelle Lernen im Hinblick auf benutzerzentrierte Aktivitäten.	1
	Das verfügbare Wissen über benutzerzentrierte Aktivitäten wird in einer praxisnahen Form angeboten.	1
	Im Rahmen eines Wissensmanagementprozesses wird der Einsatz von Methoden und Tools zur benutzerzentrierten Software-Entwicklung über einen Reviewprozess validiert.	2
	Das Schulungswesen unterstützt das organisationelle Lernen im Hinblick auf Benutzerzentrierung.	2
	Die systematische Erfassung, Verwaltung und Verfügbarmachung von usabilityrelevantem Know-how wird durch ein Tool unterstützt.	3

Tabelle 1: Anforderungen der 7 Dimensionen an jede Reifegradstufe - Fortsetzung

Dimension	Anforderungen	Reifegrad
Ausreichende Ressourcen	Die Entwicklung und Aufrechterhaltung der methodischen und physischen UE-Infrastruktur und des UE-Know-hows ist möglich.	1
	Es können externe Expertinnen hinzugezogen werden.	1
	Es wird in ausreichendem Mass interne Ausbildung durchgeführt.	2
	Es ist ein Usability Labor vorhanden.	2
	Usability Experten haben die Kapazität, in übergeordneten Gremien mitzuarbeiten.	2
	Es können kontinuierlich Anregungen zu einer Optimierung des Entwicklungsprozesses gesammelt werden.	2
	Alle Entwicklungsprojekte, bei denen eine Benutzerschnittstelle Teil des Produktes ist, können unterstützt werden.	3
	Es kann ein Tool zur Unterstützung des organisationellen Lernens entwickelt werden.	3

Tabelle 1: Anforderungen der 7 Dimensionen an jede Reifegradstufe - Fortsetzung

5.3 Zusammenfassung

Dass neben dem eigentlichen Prozessmodell zur Software-Entwicklung noch andere Faktoren eine Rolle spielen, um erfolgreich Software-Qualität zu produzieren, ist keine neue Erkenntnis. Betrachtet man die Probleme, die im Zusammenhang mit der Einführung von benutzerzentrierten Vorgehensweisen auftreten und die grossen Widerstände, die trotz eines zweifellos vorhandenen ‚Usabilitynotstands‘ anzutreffen sind, muss man zu dem Schluss kommen, dass diese anderen Faktoren jedoch in der Vergangenheit vermutlich zu wenig beachtet wurden.

Das hier erarbeitete Modell einer benutzerzentrierten Organisation betrachtet das Prozessmodell und die entsprechenden Methoden und Tools zur Umsetzung der vorgeschriebenen Aktivitäten daher zwar als wichtige Dimensionen. Diese stellen jedoch nur einen Teil der Wahrheit dar. Weitere Aspekte, wie die Organisationsstruktur, Managementaktivitäten und Kultur, Qualifikation der Mitarbeiterinnen und das Schulungswesen und Wissensmanagement werden ihnen als gleichberechtigte weitere Dimensionen hinzugefügt. Ebenso die Dimension Ressourcen, die – obschon implizit über alle anderen Dimensionen abgedeckt – zur gezielten Analyse gesondert aufgeführt wird.

Grundlage für das vorgestellte Modell liefern einerseits die vorhandenen Normen und Regelwerke zur Steigerung der *Capability* von Software-Entwicklungsprozessen und deren Pendant aus der Welt des Usability Engineerings. Um jedoch neben Prozessmodell und Methoden und Tools auch den anderen Dimensionen gerecht zu werden, wurden Erfahrungen und Probleme mit der Anwendung von Usability Engineering Prozessen ausgewertet. Anregungen vor allem im Bereich der Dimension Managementaktivitäten und Kultur stammen grösstenteils aus der Innovationstheorie.

Das Ergebnis dieses Kapitels ist ein ganzheitliches Modell einer benutzerzentrierten Entwicklungsorganisation. Konkretisiert wird das Modell durch spezifische Merkmale, die für jede der sieben Dimensionen benannt werden. Der Idee der Reifegradmodelle folgend, werden die verschiedenen Merkmale drei aufeinander aufbauenden Reifegraden zugeordnet. Dies soll nach einer erfolgten Analyse des Reifegrads einer

Organisation die Planung der nächsten strategischen Schritte auf dem Weg zur optimalen Benutzerzentrierung erleichtern.

Der eigentliche Sinn des vorgestellten Modells ist damit schon benannt: Es soll ein Rahmenwerk zur umfassenden, differenzierten Analyse einer Organisation im Hinblick auf die verschiedenen Dimensionen von Benutzerzentrierung gegeben werden, das es ermöglicht, aufbauend auf dieser Analyse die Benutzerzentrierung innerhalb der Organisation kontinuierlich durch strategische und effektive Massnahmen zu verbessern. Ziel des Modells ist es nicht, ein starres Rahmenwerk vorzugeben, mit dessen Hilfe man schwarz/weiss Entscheidungen bezüglich Benutzerzentrierung fällen könnte, wie dies eventuell im Hinblick auf eine Akkreditierung notwendig wäre.

Abbildung 14 zeigt das vollständige Reifegradmodell zur Beurteilung der Benutzerzentrierung einer Organisation.

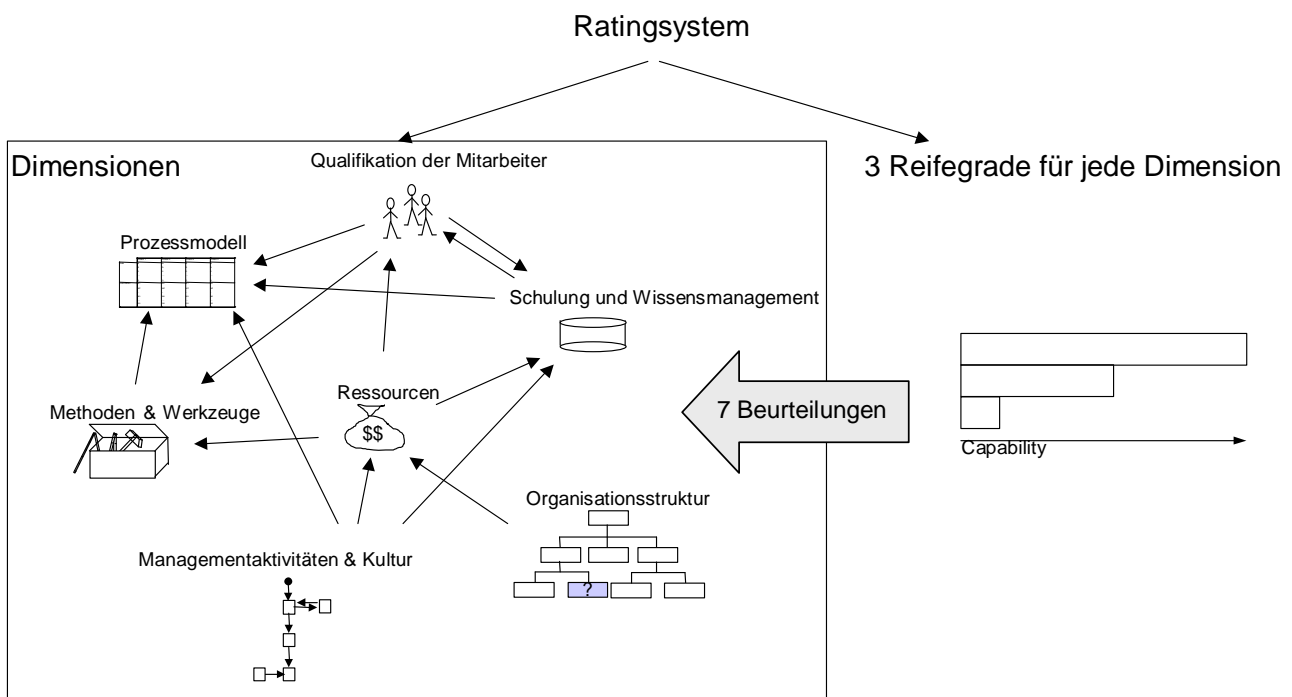


Abbildung 14: Modell zur Beurteilung von Benutzerzentrierung

6 Das Fallbeispiel UBS Schweiz

Mit der Fusion der beiden Schweizer Grossbanken, UBS Schweizerische Bankgesellschaft (Zürich) und Schweizerischer Bankverein (Basel), entstand im Juli 1998 der Konzern UBS AG. UBS Schweiz ist eine von 4 Unternehmensgruppen¹³ des UBS Konzerns (vgl. Abbildung 15), der weltweit ca. 67'000 Mitarbeiterinnen beschäftigt (Stand 31.12.2001) (UBS AG 2002). Der Geschäftsbereich IT (zukünftig GB IT genannt) von UBS Schweiz ist mit ca. 4'000 Mitarbeitern (davon knapp 2000 Entwicklerinnen) eine der grössten Software-Entwicklungsorganisation der Schweiz. GB IT ist der Geschäftseinheit Privat- und Firmenkunden von UBS Schweiz zugeordnet.

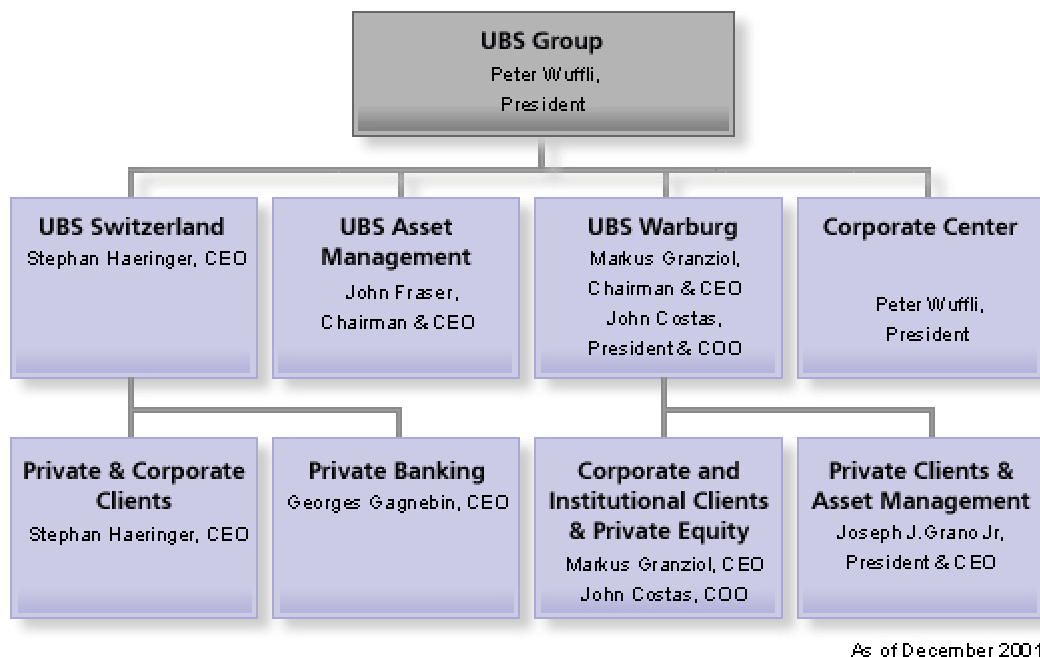


Abbildung 15: Organisationsstruktur des UBS Konzerns

Der Masterplan der GB IT Strategie von 2002 bis 2004 definiert die IT als eine Schlüsselkompetenz der Bank! Die Aufgaben beziehen sich auf die Bedienung aller IT-bezogenen Bedürfnisse von direkten Benutzerinnen sowie auf die Unterstützung der Geschäftseinheiten von UBS Schweiz und des Corporate Centers¹⁴ in deren Rolle als ‚Application owners‘ während des kompletten Applikations Lifecycles.

Ein Schwerpunkt der IT Projekte bei UBS Schweiz beschäftigt sich momentan mit der Erneuerung der gesamten Infrastruktur für die Kerndaten der Bank. Dieser Prozess (SSP = Strategic Solution Program) wird vermutlich insgesamt über ein Jahrzehnt andauern. Dabei soll das bisherige, cobolbasierte ‚Legacy-System‘ auf der Basis einer Unisys Plattform auf eine wartungsfreundliche Architektur mit modernen Plattformen, Schnittstellen und komponenten-basierten Systemen überführt werden. Zukünftig sollen nur gut eingeführte IT Industriestandards verwendet werden, Applikationen auf Nicht-Standardplattformen sollen migriert werden. (UBS IT Management 2002)

Aufgrund eines Beschlusses des IT Managements aus dem Jahr 2000 werden zukünftig alle Applikationen bei UBS Schweiz als browserbasierte Thin-Clients entwickelt. Von diesem Vorgehen verspricht man sich umfangreiche Einsparungen durch den Wegfall von Ausbreitungskosten und durch die Vereinheitlichung der IT Infrastruktur. Diese Vereinheitlichung, die sowohl Plattformen, technische Komponenten, Prozesse, Methoden als auch Entwicklungswerkzeuge umfasst, soll die Komplexität der IT-Landschaft signifikant reduzieren und damit zum einen die Stabilität und Verfügbarkeit der Applikationen erhöhen sowie die Kosten-

¹³ Seit 01.01.2002 ist zusätzlich UBS PaneWebber eine eigenständige Unternehmensgruppe.

¹⁴ Das Corporate Center ist für das gesamte Finanzmanagement des Unternehmens zuständig. Ausserdem hat es die Aufgabe, die Aktivitäten der einzelnen Unternehmensgruppen zu koordinieren. Es soll sicherstellen, dass die Unternehmensgruppen als kohärentes und wirksames Ganzes zusammenarbeiten, um die übergeordneten Ziele des Konzerns zu erreichen. Eine Aufgabe dabei ist auch die Koordination von IT Aktivitäten, vor allem im ‚e-Bereich‘.

effizienz steigern. Als Konsequenz aus diesen Entscheidungen ergibt sich, dass auch die Applikationslandschaft für die Bankmitarbeiterinnen grösstenteils erneuert wird.

Sowohl die oben genannten Ziele als auch die vorgeschlagenen Massnahmen zu ihrer Erreichung sind nahezu ausschliesslich technikzentriert. Kundeninteressen beziehen sich auf die (internen) Auftraggeber, jedoch nicht auf die Endbenutzerinnen. Benutzerinteressen werden allenfalls in sofern berücksichtigt, als Stabilität und Verfügbarkeit der Applikationen auch für Benutzerinnen von grösstem Interesse sind. Benutzerzentrierung, Benutzerzufriedenheit oder Nutzungseffizienz werden im Rahmen der IT-Strategie nicht erwähnt. Es wird zwar der Anspruch erhoben, dass die IT mit innovativen, flexiblen und führenden Lösungen das Wachstum der Bank unterstützt. Dass die Gebrauchstauglichkeit der entwickelten Applikationen für die internen und externen Kunden der Bank dabei jedoch auch einen kritischen Erfolgsfaktor darstellt, wird nicht berücksichtigt.

Um eine Vorstellung von der Komplexität der Entwicklungslandschaft bei UBS Schweiz zu vermitteln und andererseits die Ansätze und Tätigkeiten des Usability Engineerings in den entsprechenden Kontext zu stellen, werden im folgenden Kapitel zunächst die Entwicklungslandschaft von UBS Schweiz sowie die Struktur des GB IT kurz vorgestellt.

6.1 Software-Entwicklung bei UBS Schweiz

Die verschiedenen Kundengruppen von UBS Schweiz werden durch unterschiedliche Organisationseinheiten bedient. UBS Schweiz besteht aus den Organisationseinheiten Privat- und Firmenkunden und Private Banking. Für jede Kundengruppe gibt es ein eigenes Business Technology Center (BTC), das als Schnittstelle zwischen Business und IT auftritt. Bis vor kurzem gab es dadurch auch in der Applikationsinfrastruktur der Bank eine Trennung nach Kundengruppen. So konnte es durchaus vorkommen, dass verschiedene Applikationen mit nahezu identischem Funktionsumfang zur Betreuung der unterschiedlichen Kundengruppen entwickelt wurden. Vorteil dieser Aufsplittung aus Benutzerinnen- bzw. Kundenberatersicht war, dass die Applikationen – sofern die vorhandenen Entwicklungsmethoden dies ermöglicht haben – tatsächlich auf die spezifischen Bedürfnisse einer klar definierbaren Benutzergruppe zugeschnitten waren. Der Nachteil einer solchen Vorgehensweise liegt jedoch auf der Hand: Redundanzen und Unübersichtlichkeit in der Applikationslandschaft, unnötige Doppelspurigkeiten bei der Entwicklung und damit eine Vervielfachung der Entwicklungskosten.

Um diesem Problem zu begegnen, wurde im Herbst letzten Jahres innerhalb UBS Schweiz eine Matrix-Organisationsstruktur eingeführt, die die geschäftsbereichsübergreifende Koordination der IT Infrastruktur (inkl. Applikationsentwicklung) gewährleisten soll. Die sogenannte Stream-Organisation bündelt Entwicklungsaktivitäten thematisch. Ein Stream beschäftigt sich beispielsweise mit der Bereitstellung zentraler Komponenten der Vermögensverwaltung, ein anderer mit zentralen Bankprozessen. Der bezüglich benutzerzentrierter Fragestellungen interessanteste Stream ist für die Koordination aller Entwicklungsprojekte zuständig, die im Zusammenhang mit der Kundenbetreuung stehen. Hier sind z.B. alle Aktivitäten in Bezug auf Customer Relationship Management (CRM) angesiedelt, aber auch alle e-Banking Projekte. Insgesamt gibt es sechs Streams.

6.1.1 Die GB IT Value Chain

Initiatoren von IT-Projekten und damit Auftraggeber und Kunden des GB IT sind in erster Linie die beiden Geschäftseinheiten Private Banking und Privat- und Firmenkunden von UBS Schweiz (in geringem Umfang auch das Corporate Center). In der Regel werden diese von den jeweiligen Business Technology Centers (BTC) vertreten. Die Auftraggeber stellen die Verhandlungspartner für die IT Services dar, formulieren Anforderungen des Business an neue IT-Lösungen und übernehmen die Gesamtprojektleitung der IT-Projekte auf Businessseite.

Der GB IT ist in 7 Ressorts aufgeteilt, die spezifische Rollen im Rahmen eines Entwicklungsprozesses einnehmen. Die beiden Entwicklungsressorts Development Business Systems (ITBS), zu dem neu auch das Usability Engineering gehört, und Development Shared Systems (ITSD) sind für IT-Lösungen über deren gesamten Lebenszyklus verantwortlich. In der IT Value Chain (vgl. Abbildung 16) ist vorgesehen, dass diese beiden Ressorts in Person des Solution Managers als Generalunternehmer gegenüber dem Business auftreten und alle nachgelagerten Services koordinieren.

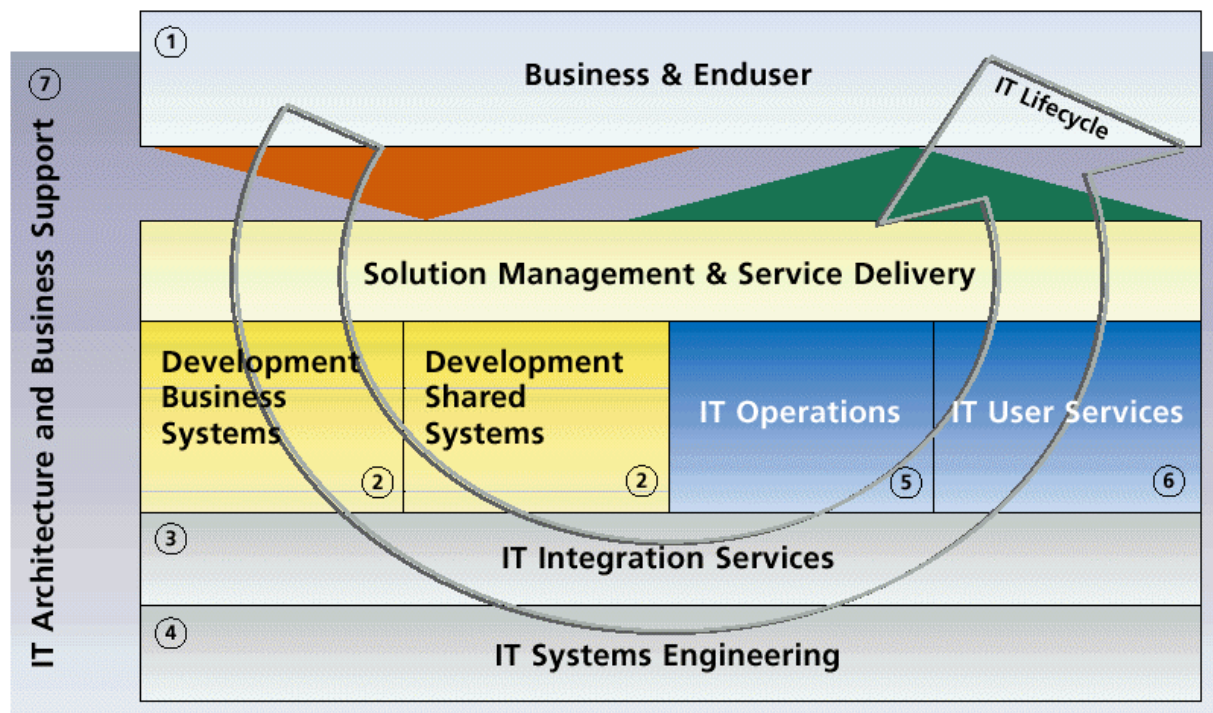
Das Ressort Integration Services (ITIN) übernimmt für die von der Entwicklung bereitgestellten Lösungen die Integrationsplanung und -koordination im Rahmen des Release- und Changemanagements und stellt Testmethoden, -werkzeuge und -daten zur Verfügung.

IT Systems Engineering (ITSE) ist verantwortlich für den Plattform-Managementprozess und hat bis vor kurzem zudem Entwicklungsservices in Form von Ausbildung und Support bereitgestellt. Die Abteilung Development Services, der auch das Usability Engineering angehörte, wurde per 01.04.2002 aufgelöst.

IT Operations (ITOS) ist für die Inbetriebnahme und den Betrieb der Applikationen, Systeme und Netze verantwortlich und gewährleistet entsprechende Serviceleistungen für Business- und Endbenutzer.

IT User Services (ITUS) erbringt flächendeckend Services für den elektronischen Arbeitsplatz und ist die Schnittstelle zwischen den Benutzerinnen und dem GB IT.

Eine übergreifende Funktion nimmt das Ressort IT Architecture and Business Support (ITAS) ein, das die Sicherheit der IT-Lösungen und Services gewährleistet und die Architekturen festlegt.



(UBS AG 2001, S. 2)

Abbildung 16: Die GB IT Value Chain

6.1.2 Usability Engineering bei UBS Schweiz

Bereits seit den 80er Jahren gab es bei der früheren Schweizerischen Bankgesellschaft (SBG) eine Gruppe, die sich mit Ergonomie beschäftigte. Zunächst lag der Fokus auf Hardwareergonomie, verschob sich zu Beginn der 90er Jahre jedoch zunehmend in Richtung Softwareergonomie. Usability Engineering war derzeit noch ein Fremdwort und zunächst nannte sich das Team ‚Fachstelle für Softwareergonomie‘. 1995 wurde bei der SBG ein stationäres Usability Labor eingerichtet, das innerhalb der Firma den Bekanntheitsgrad der Softwareergonomie-Gruppe massiv erhöhte.

Die Anzahl der Mitarbeiter bewegte sich in den letzten 20 Jahren immer zwischen 4 und maximal 10 Personen. Momentan besitzt das Usability Engineering ein Personalbudget von 600 Stellenprozent, das mit 4 Vollzeit- und 2 Teilzeitbeschäftigten nicht vollständig ausgeschöpft wird. Eine Person arbeitet auf Stundenbasis.

Die Einheit hat diverse Wechsel innerhalb der Organisationsstruktur durchgemacht. Im August 2000, zusammen mit einem Wechsel der Leitung der Fachstelle und der Bildung einer eigenständigen Sektion wurde die ‚Fachstelle für Softwareergonomie‘ in ‚Usability Engineering‘ umbenannt. Die Sektion war in der Abteilung Development Services im Ressort IT Systems Engineering (siehe Kap. 6.1.1) angesiedelt. Obwohl der Sektionsstatus durchaus als Anerkennung der Bedeutung des Usability Engineerings zu werten war, war

diese Platzierung aus verschiedenen Gründen nicht ganz glücklich. Systems Engineering innerhalb der IT Value Chain spielt eine nachgelagerte Rolle und die Abteilung Development Services stand zudem immer unter Kostendruck, da hier nicht direkt produziert wurde. Ein deutlicher Vorteil dieser Positionierung war jedoch, dass der ehemalige Abteilungsleiter sehr von den Anliegen des Usability Engineerings überzeugt war, und diese entsprechend seines Kompetenzbereiches auch nach oben kommuniziert und vertreten hat.

Bei einer erneuten Reorganisation mit Wirkung zum 01.04.2002 wurde aus Einsparungsgründen die Abteilung Development Services aufgelöst. Der Fortbestand des Usability Engineerings wurde nicht in Frage gestellt. Das Usability Engineering hat jedoch den Status einer eigenständigen Sektion verloren und gehört neu als Team einer Sektion zur Qualitätssicherung (Testing, Support, Integration und Quality Assurance, kurz TSI & QA) innerhalb einer ‚e-Entwicklungs‘-Abteilung an. Hierzu muss bemerkt werden, dass fünf der sieben IT Ressorts eine derartige Sektion besitzen und es sich daher hier nicht um eine übergeordnete Organisationseinheit handelt.

Die neue Positionierung innerhalb des Entwicklungsressorts IT Business Systems (ITBS) wird von Seiten des Usability Engineerings als vorteilhaft wahrgenommen. De facto wird sich für die Arbeit des Usability Engineerings voraussichtlich nicht viel ändern. Da jedoch der Kostendruck innerhalb der Entwicklungsressorts deutlich geringer ist, bestehen gute Chancen, dass bei einer entsprechenden Auftragslage der Personalbestand im Usability Engineering ausgebaut werden kann. Die Kehrseite dieser Medaille ist jedoch, dass zukünftig tatsächlich 95% aller Arbeiten über Projekte abgerechnet werden müssen. D.h. auch für alle Infrastrukturprojekte müssen Sponsoren oder Auftraggeber gefunden werden. Nach Aussagen des neuen Sektionsleiters sollte dies jedoch ohne grössere Probleme möglich sein.

Dass zumindest von den Aktivitäten her die neue Positionierung für das Usability Engineering passt, zeigen die folgenden Zahlen: Lässt man die Wartungsaktivitäten aussen vor, fallen bei UBS Schweiz im Jahr schätzungsweise 250 - 300 Projekte an, bei denen eine Benutzerschnittstelle entwickelt bzw. grundlegend überarbeitet wird. Ca. 200 davon werden in ITBS bearbeitet.^{15,16} Man kann also davon ausgehen, dass ein Grossteil der Applikationen mit signifikanten Benutzerzahlen (solche für Frontmitarbeiterinnen oder externe Kunden) innerhalb von ITBS entwickelt werden. Auch alle Applikationen aus dem e-Banking Bereich die durchweg als Projekte mit strategischer Bedeutung gewertet werden und seit kurzem daher einen Test im Usability Labor durchlaufen müssen, werden hier entwickelt. Mit der neuen Organisation ist das Usability Engineering in die Nähe dieser Entwicklerinnen gerückt und gehört damit zur selben Organisationseinheit, wie die sogenannten Solution Manager, die gegenüber dem Business als Generalunternehmer auftreten.

6.1.3 Relevante Einflussgrössen: Corporate Center und Marketing

Das Corporate Center befasst sich nicht nur mit zentralen Finanzmanagementaufgaben des UBS Konzerns, sondern koordiniert auch die gesamte Kommunikation der Konzerngruppen nach innen und aussen. Die Organisationseinheit ‚Web Communications & Publications‘ des Corporate Centers bemüht sich, sowohl die Publishing Infrastruktur als auch das Corporate Design zu vereinheitlichen.

Die Marketingeinheit von UBS Schweiz befasst sich klassischerweise ebenfalls mit vielen Fragestellungen, die Überschneidungen mit dem Usability Engineering aufweisen. Das Marketing ist bisher Owner eines Styleguides, der innerhalb UBS Schweiz als verbindliche Richtlinie für alle e-banking Applikationen galt und zeitweise als Styleguide für **alle** browserbasierten Applikationen (sowohl für interne als auch externe Benutzerinnen) bei UBS Schweiz gehandelt wurde. Die Bemühungen um Vereinheitlichung und Konsistenz werden von seiten des Usability Engineerings durchaus gut geheissen. Der erwähnte Styleguide orientiert sich jedoch in erster Linie an einem ansprechenden Design und kommt dabei in manchen Belangen mit den Zielen der Softwareergonomie in Konflikt. Zudem wird der Styleguide den besonderen Bedürfnissen bei der Entwicklung interner Applikationen nicht gerecht. Im Rahmen der Bemühungen des Usability Engineerings

¹⁵ Insgesamt sind bei UBS Schweiz in den beiden grossen Entwicklungsressorts im Jahr 2002 ca. 580 Projekte mit insgesamt über 700 Teilprojekten budgetiert. Die meisten Produkte mit Benutzerschnittstelle fallen in ITBS an (knapp 300), bei ITSD kann man davon ausgehen, dass nur in ca. 10 % der Projekte eine Benutzerschnittstelle entwickelt wird (ca. 30 Projekte). Das Projektbudget in den beiden Entwicklungsressorts beträgt knapp 800 Mio CHF, wovon ca. 80 % für Entwicklung bzw. Neuentwicklung und ca. 20 % für reine Wartungsprojekte budgetiert sind. (Quelle: Management Support ITBS und ITSD)

¹⁶ Jährlich werden ca. 30 – 40 Projekte vom Usability Engineering beraten. Der Umfang der Beratung variiert dabei stark, ca. 15 Produkte werden jährlich im Usability Lab getestet. Geht man pro Jahr von ca. 300 Projekten aus, in denen eine Benutzerschnittstelle entwickelt wird, so beträgt die Abdeckung mit Usability Engineering Beratung immerhin über 10 % und über 5 % aller Projekte führen einen Usability Test durch. Ein vollständiger Usability Engineering Prozess wird allerdings in keinem der Projekte durchgeführt.

um die Ownership eines neuen Styleguides, der auf der Basis softwareergonomischer Aspekte zentrale Interaktionselemente neu gestaltet, kam es in letzter Zeit häufiger zu Auseinandersetzungen um die Zuständigkeiten zwischen Marketing, Corporate Center und Usability Engineering.

6.2 Reifegrad der Benutzerzentrierung bei UBS Schweiz

Im folgenden findet für jede der im Modell einer benutzerzentrierten Entwicklungsorganisation definierten Dimensionen eine Bewertung statt. Die Reifegradstufen sind farblich codiert: weiss markiert sind die Anforderungen für den Reifegrad 1, hellgrau diejenigen für den Reifegrad 2 und grau die des Reifegrads 3.

Es hat sich gezeigt, dass es sinnvoll ist, die Bewertung ebenfalls abzustufen. In vielen Punkten wurden die Anforderungen zwar nicht vollständig erfüllt, es sind jedoch gute Ansätze zu erkennen. Um diese Informationen in der Bewertung nicht zu verlieren, wurde eine dreistufige Bewertung, angelehnt an UMM: Human Centredness Scale (Earthy 1998), durchgeführt: erfüllt (✓), teilweise erfüllt (➤), gar nicht erfüllt (✗).

Das Modell einer benutzerzentrierten Organisation gibt ein Raster für die verschiedenen Fragestellungen vor. Die Antworten auf diese Fragen sind jedoch in gewissem Masse subjektiv und entsprechen der persönlichen Sichtweise der Autorin.

6.2.1 Die Entwicklungsrichtlinien (ERL) von UBS Schweiz als Standardprozessmodell

Kommentar	Bewertung
<p>Mit der geringfügigen Überarbeitung der Entwicklungsrichtlinien von UBS Schweiz im Herbst 2001 (ERL 3.2) wurden einige wichtige Aspekte der Benutzerzentrierung in den Standardentwicklungsprozess integriert. Neu wird beispielsweise eine Definition der Benutzergruppen im Rahmen des Business-Case vorgeschlagen.</p> <p>Es gibt keine eigene Kategorie ‚Benutzerzentriertheit‘ und dementsprechend auch keine Verantwortlichkeiten. Bisher sind die Benutzeraspekte nicht im Rollenkonzept abgedeckt. Im nächsten Release (ERL 4.0) soll es jedoch eine Rolle ‚Usability Engineer‘ geben. Neu wird in den ERL 4.0 auch eine Spezifikation des konzeptuellen Modells der Benutzerschnittstelle im Rahmen der Anforderungsspezifikation integriert. Ausserdem wird die Entwicklung eines Prototypen empfohlen. Das Usability Engineering wird auch in Zukunft keine Sign-off relevante Fachstelle werden, es wird jedoch bereits heute an verschiedenen Stellen in den ERL darauf verwiesen. Eine kontinuierliche Erweiterung der ERL um benutzerzentrierte Aspekte ist vorgesehen. Es muss jedoch gewährleistet sein, dass alle verbindlich vorgeschriebenen Aktivitäten auch bedient werden können.</p>	➤
<p>Das Prozessmodell ist ergebnisorientiert. Es schreibt keine Methoden zur Erreichung der Ergebnisse vor. In den Templates für die Ergebnisse befinden sich Verweise zu den jeweiligen Fachstellen, die zur Unterstützung bei der Erstellung hinzugezogen werden können bzw. Hinweise, wie ein Ergebnis ggf. erstellt werden kann.</p>	✓

Tabelle 2: Reifegrad des Prozessmodells von UBS Schweiz hinsichtlich Benutzerzentrierung

Kommentar	Bewertung
Eine Evaluation der Usability des Endproduktes ist für Projekte mit Endkunden der Bank als Zielgruppe am Ende des Entwicklungsprozesses vorgeschrieben. Für die Mehrzahl der Projekte mit internen Zielgruppen ist die Usability Evaluation freiwillig, die Möglichkeit eines Usability Tests wird jedoch erwähnt.	➤
Das vorhandene Prozessmodell unterstützt in begrenztem Umfang Iteration. Die Darstellung ist jedoch eher linear. Das Modell ist dokumentenzentriert. Vorgaben für die Erstellung der Dokumente sollen diese erleichtern.	➤
Software-Engineering und Usability Engineering Aktivitäten sind integriert.	✓
Die Dokumentenorientierung wird teilweise zur Dokumentenlast. Nicht selten werden Dokumente erstellt, die mehrere hundert Seiten umfassen. Diese können dann weder konsequent aktualisiert werden, noch sind sie geeignet, um sie tatsächlich als Arbeitsdokumente im Prozess zu verwenden.	✗
Der Prozess endet mit dem Production Sign-off, der das Produkt für die Ausbreitung freigibt.	✗
Die Projektdokumentation beinhaltet implizit auch Aspekte der Benutzerzentrierung. Diese werden jedoch von den Projekten sehr unterschiedlich ausgelegt. Eine systematische Durchführung von Usability Engineering Aufgaben findet nicht statt, wodurch auch keine Wissensdokumentation möglich ist.	➤
Das Prozessmodell ist für verschiedene Projekttypen und –größen theoretisch skalierbar. Praktisch wird eine Skalierung jedoch aufgrund der Dokumentenzentriertheit kaum unterstützt (auch kleinere Projekte fühlen sich zum Teil aufgefordert, sämtliche Dokumente auszuarbeiten).	➤
Das Prozessmodell präsentiert sich grundsätzlich gebrauchstauglich, ist jedoch durch die grosse Anzahl vorhandener Ergebnisdokumente (52), durch die grosse Anzahl der Fachstellen (10) und Rollen (20) aus Business und IT in der Gebrauchstauglichkeit eingeschränkt. Die zur Zeit durchgeführte Überarbeitung der ERL soll hier Abhilfe schaffen. Es ist jedoch geplant, trotz Reduktion des Gesamtumfangs, weitere benutzerzentrierte Aspekte zu integrieren und auch eine Rolle ‚Usability Engineer‘ einzufügen.	➤

Tabelle 2: Reifegrad des Prozessmodells von UBS Schweiz hinsichtlich Benutzerzentrierung - Fortsetzung

6.2.2 Das Angebot an Methoden und Tools zur benutzerzentrierten Software-Entwicklung

Kommentar	Bewertung
<p>Lange Zeit lag der Schwerpunkt des Usability Engineerings bei UBS Schweiz auf der Durchführung von Usability Tests und anschliessenden Redesignvorschlägen. Im Jahr 2001 wurde jedoch ein Methodenhandbuch erstellt, das neben dem klassischen Usability Testing die wichtigsten frühen Methoden des Usability Engineerings, wie bspw. Benutzergruppenanalyse, Kontext- und Aufgabenanalyse und Prototyping, umfasst. Zudem wird momentan ein Styleguide erstellt, der für alle browserbasierten Applikationen von UBS Schweiz (mit interner und externer Nutzung) verbindlich werden soll. Alle im Prozessmodell definierten Teilergebnisse mit Benutzerbezug können durch das Methodenangebot abgedeckt werden.</p> <p>Tools zur Unterstützung benutzerzentrierter Methoden sind vorhanden. Für das Requirementsmanagement steht theoretisch Caliber RM zur Verfügung. Die Lizenznehmer von Caliber RM beklagen sich jedoch über die mangelhafte Unterstützung und die schlechte Pflege des Tools, weshalb die Nachfrage stark zurückgegangen ist.</p> <p>Im Zusammenhang mit Prototypingaktivitäten kommen einerseits PowerPoint und Visio zum Einsatz (LoFi-Prototypen), es werden jedoch auch HTML Prototypen erstellt (z.B. mit Dreamweaver). Der Einsatz von Visio wird dadurch gefördert, dass Standardinteraktionselemente vorgegeben sind, die dem UBS Application Styleguide entsprechen.</p>	✓
<p>Die Auswahl geeigneter Methoden für das jeweilige Projekt wird durch entsprechendes Consulting unterstützt. Die Methodenanwendung wird grösstenteils von den Projektrahmenbedingungen geprägt. Die Projekte werden angehalten, hierzu einen Usability Experten hinzuzuziehen.</p>	✓
<p>Die vorhandenen Methoden befinden sich teilweise noch in der Evaluationsphase (vor allem Kontext- und Aufgabenanalyse); teilweise sind jedoch auch umfangreiche Erfahrungen mit den Methoden vorhanden (z.B. Papier-Prototyping). Bei der Ausarbeitung der Methoden wurde Wert darauf gelegt, dass diese den Projekten als Pakete angeboten werden können. Dies kommt der Realität der internen Verrechnung entgegen und reduziert bei den Projekten die Hemmschwelle, das Angebot des Usability Engineerings in Anspruch zu nehmen.</p>	➔

Tabelle 3: Qualität des Methoden- und Toolangebots

6.2.3 Benutzerzentrierung und Organisationsstruktur

Kommentar	Bewertung
Die Mitarbeiter des Usability Engineerings arbeiten einerseits direkt in den Projekten mit, andererseits erfüllen sie jedoch auch reine Beratungsfunktionen, führen Schulungen durch oder unterstützen einzelne Aktivitäten.	✓
Das Usability Engineering ist eine eigenständige Organisationseinheit und auch als solche wahrnehmbar. Dies wird vor allem durch die auffälligen Räumlichkeiten mit dem Usability Labor unterstützt. Seit Anfang April 2002 ist das Usability Engineering keine eigene Sektion mehr, sondern wird als Team in einer Qualitätsmanagementsektion innerhalb eines Entwicklungsressorts geführt. Für die eigentliche Arbeit und die Wahrnehmung von aussen ändert sich dadurch jedoch praktisch nichts.	✓
Das Usability Engineering hat die Freiheit, sowohl auf Anfragen des Business, der BTCs als auch der IT zu reagieren. Diese Vorgehensweise war nicht zu jeder Zeit problemlos möglich, da eine Verrechnung der Serviceleistungen für das Business/BTC direkt mit dem Usability Engineering in der ‚IT Value Chain‘ nicht vorgesehen ist.	✓
<p>Je nach Projekt ist die Entscheidungsbefugnis über die Ausgestaltung der Benutzerschnittstelle unterschiedlich verteilt. Das letzte Wort sprechen in der Regel die Entwickler, die das GUI realisieren. Bezüglich der Gestaltung sind Richtlinien in Form von Styleguides vorgegeben, die bis anhin vom Marketing vorgelegt wurden. Wichtige Rahmenbedingungen liefern auch die technischen Vorgaben der GB IT Webplattform, die vorschreiben, dass <i>Classic GUIs</i> entwickelt werden sollen, also solche, die mit purem HTML und minimalem JavaScript realisierbar sind, um Browserunabhängigkeit zu garantieren (im Gegensatz zu <i>Rich GUIs</i>, die auch die Verwendung von Java oder vermehrten Einsatz von JavaScript erlauben, von externen Benutzerinnen jedoch erfordern würden, bestimmte Applets, Plugins oder gar andere Browsertypen zu installieren oder zumindest JavaScript zu aktivieren). Da vor allem auf Kundenberaterebene verschiedene Applikationen sowohl für den internen Gebrauch als auch für den externen Gebrauch (beim Kunden) vorgesehen sind, bestehen Bestrebungen, für alle Applikationen den kleinsten gemeinsamen technologischen Nenner <i>Classic GUIs</i> zu suchen.</p> <p>In manchen Projekten wird die Entwicklung des GUIs und damit auch die Entscheidungsbefugnis an eine externe Firma vergeben.</p>	✗
In Projekten, wo das Usability Engineering früh involviert ist, erhält der Usability Experte mindestens Beratungsfunktion, z.T. sogar umfassende Entscheidungsbefugnis bezüglich Konzept und Gestaltung des GUIs. In der Regel werden in solchen Projekten gemeinsam mit Business- und IT-Vertreterinnen Lösungen erarbeitet.	➤

Tabelle 4: Unterstützung der Benutzerzentrierung durch eine geeignete Organisationsstruktur

6.2.4 Managementaktivitäten und Kultur zur Förderung von Benutzerzentrierung

Kommentar	Bewertung
<p>Die Einstellung der Entwicklungsteams gegenüber Fragen der Usability sind unterschiedlich. In der Regel wird dieses Thema durchaus ernst genommen und vor allem von jüngeren Entwicklerinnen auch als sehr spannend und erfolgskritisch bewertet. Das Bewusstsein, dass bezüglich der Usability eines Produktes Probleme auftauchen können, ist vorhanden. Die Arbeitsweise von Entwicklern ist jedoch zum Teil bezüglich der Lösung von Usabilityproblemen eher hinderlich. Usabilityprobleme auf konzeptioneller Ebene lassen sich im allgemeinen nicht durch die Korrektur von ein paar Zeilen Code beheben, sondern erfordern eine sorgfältige Überarbeitung des Konzeptes. Ein Problembewusstsein ist zwar vorhanden, das Bewusstsein der adäquaten Lösungswege ist jedoch noch unterentwickelt.</p>	➤
<p>Die Wertsetzungen des IT Managements sind technikorientiert. Die wahrgenommenen Probleme beziehen sich zur Zeit vor allem auf die grundlegende technische Infrastruktur. Die IT versteht sich als Konstrukteurin und Bewahrerin dieser Infrastruktur, die zweifelsohne das Herzstück der Bank darstellt. Zu dieser Infrastruktur gehören selbstverständlich auch alle Applikationen, die für das Bankgeschäft und die Administration der Bank benötigt werden. Bei den Applikationen wird der Schwerpunkt darauf gelegt, dass diese jetzt und in Zukunft lauffähig und stabil zur Verfügung stehen und die Pflegekosten sich in einem vertretbaren Rahmen halten. Benutzerzufriedenheit wird hin und wieder als Schlagwort gebraucht, jedoch sehr randständig thematisiert.</p>	X
<p>Die Einstellung gegenüber Innovation ist eher zurückhaltend. Die Bank kann sich aufgrund der Sensitivität der vorhandenen Daten und Prozesse nur auf bewährte Technologien stützen, deren Leistungsfähigkeit erprobt und nachgewiesen ist. Erschwerend für Innovationen kommt hinzu, dass jede Form von Veränderung aufgrund der schieren Grösse des GB IT im besten Falle nur träge umzusetzen ist.</p> <p>Was für die generelle Kultur von UBS Schweiz gilt, gilt in hohem Masse auch für den Entwicklungsprozess. Von Veränderungen der Entwicklungsrichtlinien sind theoretisch so viele Stellen betroffen, dass hier besondere Zurückhaltung geübt wird. Allein ein erkannter Bedarf an Methoden, Werkzeugen oder Know-how reicht nicht aus, um eine Innovation diesbezüglich durchzusetzen. Zudem besteht kaum eine Nachfrage nach einem benutzerzentrierten Entwicklungsprozess. Sämtliche Initiativen hierfür müssen daher vom Usability Engineering ausgehen.</p>	X
<p>Es gibt keine Massnahmen um die internen bzw. externen Benutzerinnen der bei UBS Schweiz produzierten Software dazu zu ermutigen, Ansprüche an die Usability der Software zu formulieren oder Probleme mit der Usability zu kommunizieren.</p>	X
<p>Benutzerzentrierung ist kein Thema, das systematisch gefördert wird.</p>	X
<p>Entwicklungsprojekte sind dann erfolgreich abgeschlossen, wenn der vorgegebene Zeit- und Kostenrahmen nicht massgeblich überschritten wird. Eine Erfolgskontrolle im Hinblick auf Benutzerzufriedenheit in der Nutzungsphase findet nicht statt.</p>	X

Tabelle 5: Eignung von Managementaktivitäten und Kultur zur Förderung der Benutzerzentrierung

Kommentar	Bewertung
Es bestehen keine spezifischen Anreize zur Förderung von Benutzerzentriertheit im Software-Entwicklungsprozess.	X
Zwar ist es auf Seiten des Business / BTC durchaus üblich, bei der Entwicklung von Applikationen für interne Benutzer Endbenutzerinnen einzubeziehen. Die Sichtweise, dass Software entwickelt wird, um Bedürfnisse von Benutzerinnen zu befriedigen, ist innerhalb der IT jedoch nicht verbreitet. Durch die Trennung von Zuständigkeiten in Business / BTC (Anforderungsspezifikationen) und IT wird diese Situation noch verschärft.	↗
Weder beim Management, noch beim Business, noch bei den Entwicklern besteht das Bewusstsein von der umfassenden Bedeutung von Usability. Es wird teilweise erkannt, dass schlechte Performanz von browserbasierten Systemen für die fehlende Akzeptanz auf der Seite der Benutzerinnen verantwortlich ist. Lösungsansätze für dieses Problem sind rein technikzentriert und schliessen eine Optimierung auf Interaktionsebene nicht mit ein. Ebenso fehlt auf Benutzerseite ein Bewusstsein für das Thema Usability. Die Benutzerinnen selbst denken in hohem Masse an Funktionalität, die sie gerne hätten. Zudem ist Sicherheit und Performanz für sie eine relevante Grösse. Ansonsten scheint eine gewisse Abhärtung gegenüber unergonomischen und inkonsistenten Benutzerschnittstellen vorzuherrschen. Werden Benutzer in die Anforderungsphase involviert, haben sie den Anspruch, dass alle ihre persönlichen Bedürfnisse durch die neue Applikation befriedigt werden.	X
Führt eine unzureichende Benutzerzentrierung zu einer unnötigen Verteuerung oder Verlängerung oder gar zum Abbruch von Entwicklungsprozessen, so wird dies weder als Problem mangelnder Benutzerzentrierung adressiert, noch werden konkrete Sanktionen ausgesprochen, die zukünftig die Benutzerzentrierung in den Vordergrund rücken würden.	X
Die IT versteht sich einerseits als Auftragnehmerin gegenüber dem Business, das die Applikationen spezifiziert. Gleichzeitig sieht sie sich als Hüterin der vorhandenen Infrastruktur und diktiert dem Business ein Rahmenwerk innerhalb dessen die Lösungen angesiedelt sein müssen. Ein direkter Kontakt zwischen IT-Entwicklerinnen und Benutzern findet kaum statt. Kommt es zu Zielkonflikten zwischen IT und Business setzt sich die IT in der Regel durch, da innerhalb der Bank ein Abhängigkeitsverhältnis zwischen Auftraggeber und Auftragnehmer besteht, das nicht einer realen Marktsituation entspricht. Einige Projekte reagieren darauf, indem sie die UBS IT bei der Entwicklung umgehen, wodurch jedoch andererseits das Problem auftritt, dass Softwareprodukte erstellt werden, die die hausinterne IT später nicht pflegen kann oder will. Gemeinsame Zielsetzungen sind nicht erkennbar.	X

Tabelle 5: Eignung von Managementaktivitäten und Kultur zur Förderung der Benutzerzentrierung
Fortsetzung

6.2.5 Die Qualifikation der Mitarbeiterinnen im Hinblick auf benutzerzentrierte Aktivitäten

Kommentar	Bewertung
<p>Die im Usability Engineering vorhandenen Erfahrungen mit frühen Aktivitäten eines benutzerzentrierten Entwicklungsprozesses, wie beispielsweise eine Kontextaufgabenanalyse, sind noch gering. Der Erfahrungsschatz wird nach und nach von hinten nach vorne im Prozess aufgebaut. So bestehen jahrelange Erfahrungen im Bereich des Usability Testings und des Interaktionsdesigns und Prototypings. Es gab jedoch noch wenig Gelegenheit, Methoden im Bereich der Anforderungsspezifikation zu erproben und Sachkenntnis hierüber zu erlangen.</p>	➤
<p>Die Mitarbeiterinnen des Usability Engineerings sind grösstenteils Mitglieder des ACM bzw. der Swiss CHI und suchen regelmässig den Austausch mit anderen Fachleuten aus dem Gebiet der Mensch-Computer-Interaktion.</p>	✓
<p>Ausserhalb des Usability Engineerings gibt es keine Stellenbeschreibung, die die Kenntnisse bei der Durchführung benutzerzentrierter Aktivitäten verlangen würde. Entsprechend sind auch keine Leistungsnachweise oder Fortbildungsnachweise vorgesehen.</p>	✗
<p>Im Rahmen eines Entwicklungsprozesses bei UBS Schweiz kommen theoretisch verschiedenste Akteure mit der Ausführung benutzerzentrierter Aktivitäten zum Zug. Mitarbeiterinnen des Business haben häufig selbst ‚Fronterfahrung‘ und sind daher in hohem Masse geeignet, um mit Benutzern zu kommunizieren. Allerdings fehlt dort häufig jegliche Art von Ausbildung über die adäquate Einbeziehung von Benutzerinnen in den Spezifikationsprozess. Stattdessen wird häufig von den eigenen Erfahrungen und Arbeitsweisen extrapoliert, was insofern problematisch ist, da diese zum Teil bereits einige Zeit zurückliegen und die Verschiedenheit von Benutzergruppen zu wenig wahrgenommen wird.</p> <p>Businessanalysten und BTC Mitarbeiterinnen haben häufig einen Wirtschaftsinformatikhintergrund. Ihnen ist das Thema Usability vermutlich grösstenteils einmal begegnet, allerdings eher randständig. Tatsächlich ist für viele dieser Mitarbeiter die Einbeziehung von Benutzerinnen selbstverständlich, obwohl auch hier z.T. deutliche Mängel in der Qualifikation bezüglich Benutzereinbeziehung erkennbar sind.</p> <p>Die Mitarbeiter des Usability Engineerings besitzen entweder langjährige Erfahrungen im Interaktionsdesign und/oder haben eine entsprechende Ausbildung (vorwiegend in Psychologie) genossen. Die Qualifikationen liegen grösstenteils auf der Ebene der Softwareergonomie. Explizite Qualifikationen bezüglich früher benutzerzentrierter Vorgehensweisen sind noch im Aufbau begriffen.</p>	➤

Tabelle 6: Stand der Qualifikation der Mitarbeiter

6.2.6 Unterstützung von Benutzerzentrierung durch Schulungswesen und Wissensmanagement

Kommentar	Bewertung
In unregelmässigen Abständen werden Schulungen angeboten, bei denen vor allem Entwicklerinnen bezüglich Softwareergonomie und Interaktionsdesign geschult werden. Indirekt wird hierbei auch das Thema Benutzerzentrierung angesprochen. Eine systematische Aus- und Weiterbildung für die Durchführung von benutzerzentrierten Aktivitäten besteht nicht.	➤
Die sogenannten ‚GUI Guidelines‘, die im Intranet von UBS Schweiz publiziert sind, bieten eine gute Wissensquelle für die Grundlagen der Softwareergonomie. Es gibt kein vergleichbares Angebot, das in die Grundzüge benutzerzentrierter Entwicklungsprozesse einführen würde. In geringem Umfang bietet die Homepage des Usability Engineerings Informationen zu diesem Thema.	➤
Es gibt keinen Prozess, der es ermöglichen würde, die Erfahrungen mit bestimmten Methoden und Tools zu sammeln, zu reviewen und zur Verbesserung der Methoden und Tools rückzuspeisen.	X
Zur Förderung des Bewusstseins bezüglich Usability und Benutzerzentrierung organisationsweit existiert kein Schulungskonzept.	X
Es ist kein Tool vorhanden, das das Wissensmanagement bezüglich Benutzerzentrierung unterstützen würde.	X

Tabelle 7: Reifegrad des Schulungswesens und Wissensmanagements bezüglich Benutzerzentrierung

6.2.7 Ressourcen zur Förderung von benutzerzentrierter Software-Entwicklung

Kommentar	Bewertung
Es gibt Vorgaben, dass 95% der Arbeitszeit innerhalb der IT auf Projekte gebucht werden müssen. Dies wird zukünftig auch für das Usability Engineering gelten. Dies bedeutet, dass Infrastrukturaktivitäten, die ein wichtiger Bestandteil zur Förderung von Innovation sind, nicht mehr ohne weiteres bedarfsgemäss durchgeführt werden können, sondern in Form von internen budgetierten Projekten abgewickelt werden müssen. Die Abhängigkeit von der Gunst des Sektions- und Abteilungsleiters, die inhaltlich weit von den Themen des Usability Engineerings entfernt sind, ist vermutlich problematisch.	➤
Zur Zeit wird ein Styleguide für browserbasierte Applikationen entwickelt, für den das Usability Engineering die Ownership besitzt. Für dieses Projekt sind umfangreiche Ressourcen gesprochen worden, die es ermöglichen, dass alle Mitarbeiterinnen des Usability Engineerings involviert werden können. Gleichzeitig ist dieses Projekt auch geeignet, um sehr viel Wissen bezüglich Softwareergonomie zu generieren und zu transferieren.	

Tabelle 8: Ressourcenverfügbarkeit zur Förderung von Benutzerzentrierung

Kommentar	Bewertung
Die Nachfrage nach Usability Engineering Unterstützung ist teilweise grösser als die vorhandene Kapazität. In diesen Fällen ist es möglich, dass externe Expertinnen vermittelt werden. Die Hinzuziehung externer Experten zum gezielten Aufbau von Know-how in Bezug auf benutzerzentrierte Methoden ist unter den gegebenen Abrechnungsbedingungen schwierig.	➤
Zwei Personen im Usability Engineering Team sind explizit auch mit Schulungsaufgaben betraut. Allerdings bestehen hier deutliche Ressourcenkonflikte, so dass Schulungsaktivitäten allenfalls sporadisch durchgeführt werden können. Um in ausreichendem Masse in diesem Gebiet tätig werden zu können, müssten die Ressourcen hier deutlich aufgestockt werden.	➤
Bei UBS Schweiz gibt es seit 1995 ein stationäres Usability Labor, dessen Erhaltung nicht in Frage gestellt wird. Allerdings gibt es Bestrebungen, die Auslastung des Labors zu erhöhen, was angesichts der Bemühungen im Bereich der frühen Methoden zu Ressourcenkonflikten innerhalb des Usability Engineering Teams führt.	✓
Die Mitarbeiterinnen des Usability Engineerings sind teilweise in Gremien aktiv. Diese Tätigkeiten werden jedoch nicht als Arbeitszeit angerechnet und nicht von UBS Schweiz unterstützt.	✗
Seit 1999 gibt es bei UBS Schweiz einen definierten Standardentwicklungsprozess (Entwicklungsrichtlinien ERL). Zur Unterstützung dieses Prozesses werden Schulungen und Consulting angeboten. Aufgrund der Erfahrungen mit dem Prozess, die gewisse Mängel aufgezeigt haben, haben die Verantwortlichen eine Überarbeitung der ERL in die Wege geleitet. Leider sind in den letzten Jahren die Mittel für die Pflege der ERL immer weiter reduziert worden, zudem fand sich lange Zeit kein Auftraggeber für einen komplett neuen Release der ERL. Die Einbindung von benutzerzentrierten Aktivitäten in die ERL ist noch unvollständig. Für die systematische Sammlung von Erfahrungen mit benutzerzentrierten Prozessen fehlt bisher die Möglichkeit.	➤
Das Verhältnis zwischen Usability Engineering Expertinnen zur Anzahl laufender Entwicklungsprojekte erlaubt es nicht, alle relevanten Projekte zu unterstützen.	✗
Organisationelles Lernen in Bezug auf benutzerzentrierte Entwicklungsprozesse ist zur Zeit kein Thema bei UBS Schweiz. Daher bestehen weder die Bestrebungen noch das Budget, um ein Tool zu diesem Zweck zu entwickeln.	✗

Tabelle 8: Ressourcenverfügbarkeit zur Förderung von Benutzerzentrierung - Fortsetzung

6.2.8 Zusammenfassende Bewertung der Benutzerzentriertheit von UBS Schweiz

Dimension	Reife- stufe	Zusammenfassung / Begründung
Prozessmodell	1	Mit den Entwicklungsrichtlinien besitzt UBS Schweiz ein gut strukturiertes, flexibles, methodenunabhängiges und prinzipiell gut an das jeweilige Projekt anpassbares Entwicklungsprozessmodell. Wie erste Erfahrungen gezeigt haben, ist es problemlos möglich, auch benutzerzentrierte Aktivitäten in dieses Modell zu integrieren. Die Ausformulierung dieser Aktivitäten und eine entsprechende Erfolgskontrolle fehlen jedoch noch weitestgehend. Auch die praktische Anwendbarkeit des Prozessmodells sollte noch verbessert werden. Das Dilemma zwischen Vereinfachung des Prozessmodells (z.B. durch eine Reduktion der Ergebnisse, Rollen und Fachstellen) und einer Integration von benutzerspezifischen Funktionen muss gelöst werden. Dies kann nur dann zugunsten der Benutzerzentrierung geschehen, wenn diese auch zu einem anerkannten Wert oder sogar Schwerpunkt der Entwicklung gemacht wird.
Methoden- und Toolangebot	2	Das Methodenangebot auf der Seite des Usability Engineerings ist grundsätzlich geeignet, um einen Usability Engineering Prozess vollständig abzudecken. Zum Teil fehlt es jedoch an Erfahrung mit der Durchführung der Methoden, so dass nicht davon ausgegangen werden kann, dass alle Methoden bereits ausgiebig evaluiert und optimiert sind. Spezielle Tools zur Unterstützung der benutzerzentrierten Methoden, wie Requirements Management und Prototypingtools, kommen in mässigem Umfang zur Anwendung. Der Einsatz von Tools sollte weiter gefördert werden.
Organisationsstruktur	2	Die Organisationsstruktur des GB IT bietet zwar nicht die Möglichkeit einer Stabsfunktion, wie sie das Usability Engineering darstellen könnte. Allerdings stellt das Usability Engineering eine eigenständige Einheit dar (daran ändert sich auch nichts durch den verlorenen Sektionsstatus) und hat dadurch die Möglichkeit, sowohl zentrale Funktionen wahrzunehmen als auch Projektarbeit zu verrichten. Die Struktur ist zudem auch über die Abteilungs- oder Ressortgrenzen hinaus durchlässig, so dass diesbezüglich keine Einschränkung besteht, welche Projekte unterstützt werden können. Auf der Ebene der Projektorganisation fehlt es den Usability Fachleuten jedoch noch an Durchsetzungskraft und Entscheidungsbefugnis.

Tabelle 9: Zusammenfassende Bewertung der Benutzerzentriertheit von UBS Schweiz

Dimension	Reife- stufe	Zusammenfassung / Begründung
Managementaktivitäten und Kultur	0 - 1	<p>Die Kultur bei UBS Schweiz und die sichtbaren Managementaktivitäten im Hinblick auf Benutzerzentrierung sind stark unterentwickelt. Benutzerzentrierung wird innerhalb dieser übergeordneten Ebene überhaupt nicht thematisiert und es ist auch davon auszugehen, dass kein weitreichendes Bewusstsein bezüglich der Problematik vorhanden ist.</p> <p>Auch sind die vorhandenen Werthaltungen nicht unbedingt dazu geeignet, Innovationen im Bereich der IT zu fördern.</p> <p>Den Benutzerinnen wird zwar häufig die Möglichkeit gegeben, Mängel zu formulieren, Benutzerfeedback wird jedoch nicht systematisch eingeholt. Wird ein Produkt nicht in dem Masse genutzt, wie dies vorgesehen war, so hat das keinerlei Auswirkungen auf die Erfolgsbeurteilung des Entwicklungsprojektes (weder bei IT noch Business).</p> <p>Häufig ist die Situation aber auch so, dass Benutzerinnen gar keine Wahl haben, ob sie ein Produkt verwenden wollen oder nicht. Fehlende Benutzerzentrierung als grundlegendes Problem vieler Entwicklungsprozesse wird nicht als solches erkannt. Zielkonflikte zwischen IT und Business werden durch die organisationelle Trennung der Bereiche auch innerhalb von Entwicklungsprojekten und durch eine gewisse Vormachtstellung der IT aufgrund ihrer technischen Monopolstellung häufig zu Ungunsten der Benutzer entschieden.</p>
Mitarbeiterinnen Qualifikation	1	<p>Die Qualifikation der Mitarbeiterinnen des Usability Engineerings repräsentiert über weite Strecken die Erfahrungen in diesem Bereich in der Schweiz. In Zukunft müssen sicherlich vermehrt Anstrengungen unternommen werden, weiteres Know-how im Bereich der frühen benutzerzentrierten Methoden zu erwerben, was unter anderem auch durch den regelmässigen Austausch mit externen Usability Fachleuten und den Besuch von Konferenzen usw. gefördert wird.</p> <p>Ungünstiger sieht die Qualifikation und Fortbildung bezüglich Benutzerzentrierung jedoch in anderen Gebieten aus. Weder in der IT noch in den Business- /BTC Einheiten existieren diesbezüglich Anforderungen in den Stellenbeschreibungen und entsprechend ist auch wenig explizites Know-how vorhanden.</p>
Schulungswesen und Wissensmanagement	0 - 1	<p>Das Schulungswesen befasst noch viel zu wenig mit dem Thema Benutzerzentrierung. Ausschliesslich softwareergonomische Fortbildungen werden in geringem Umfang angeboten. Hier besteht ein grosser Nachholbedarf sowohl inhaltlich als auch konzeptionell, um einerseits das Thema Benutzerzentriertheit voranzutreiben und andererseits auch das entsprechende Know-how zur Anwendung von benutzerzentrierten Methoden zu vermitteln. Ein systematischer Wissensmanagementprozess zur Generierung und Weiterentwicklung von Know-how in diesem Gebiet fehlt.</p>

Tabelle 9: Zusammenfassende Bewertung der Benutzerzentriertheit von UBS Schweiz - Fortsetzung

Dimension	Reife- stufe	Zusammenfassung / Begründung
Ressourcen	1	Angesichts der Grösse des GB IT (4000 Mitarbeiterinnen) im Vergleich zur kleinen Usability Engineering Einheit (6 Mitarbeiter) ist es offensichtlich, dass die Kapazitäten für die theoretisch notwendigen Infrastruktur-, Consulting-, Schulungs- und Koordinationsaufgaben nicht ausreichend sind. Ein Wachstum der Usability Engineering Einheit wird jedoch nur im Zuge einer steigenden Nachfrage akzeptiert. Dies steht zum Teil damit in Konflikt, dass für eine steigende Nachfrage zunächst einige der vorgehend besprochenen Schwachstellen behoben werden müssten, was ohne eine Steigerung der Ressourcen nur langsam, wenn überhaupt möglich ist.

Tabelle 9: Zusammenfassende Bewertung der Benutzerzentriertheit von UBS Schweiz - Fortsetzung

6.3 Zusammenfassung

Der Stand der Benutzerzentrierung bei UBS Schweiz ist hinsichtlich der verschiedenen Dimensionen des vorgestellten Reifegradmodells sehr heterogen. Die Dimensionen Prozessmodell, Methoden- und Tools, Organisationsstruktur sowie Ressourcen erreichen zwar nicht den höchsten denkbaren Reifegrad, es wird jedoch deutlich, dass die bisherigen Bemühungen in diesen Bereichen teilweise bereits gefruchtet haben und dass sich eine positive Entwicklung abzeichnet. Interessanterweise scheinen sich diese Bereiche auch parallel zu einander zu entwickeln. Eine Erweiterung der Entwicklungsrichtlinie von UBS Schweiz um benutzerzentrierte Aspekte ist absehbar und bei den Verantwortlichen ist das Bewusstsein über die Bedeutung der Benutzerzentrierung im Entwicklungsprozess vorhanden. Offensichtliche Schwachstellen im Sinne der vorhergehenden Analyse sind vor allem im Bereich der Managementaktivitäten und Kultur zur Förderung der Benutzerzentrierung und im Schulungswesen und Wissensmanagement – und damit einhergehend auch bei der Qualifikation der Mitarbeiterinnen – erkennbar.

Die Aufteilung der Zuständigkeiten bei der Software-Entwicklung auf Organisationseinheiten des Business, BTC und IT erfordert gezielte Massnahmen unter Berücksichtigung der jeweiligen Charakteristika und spezifischen Aufgaben dieser Einheiten.

Die Quasi-Monopolstellung der IT gegenüber den Auftraggebern, führt teilweise dazu, dass sich die IT nicht nur als Dienstleisterin, sondern auch als Hüterin des heiligen Grals – der technischen Infrastruktur – sieht. Dieser besonderen Rolle der IT muss bei allen Aktivitäten zur Steigerung der Benutzerzentrierung Rechnung getragen werden, was konkret bedeutet, mit höchster Priorität Management Buy-in innerhalb der IT zu suchen.

Im Umgang mit den BTC / Businesseinheiten muss sorgfältig darauf geachtet werden, dass es nicht zu Kompetenzstreitigkeiten kommt. Vergleichbares gilt auch für die Anliegen des Marketings und des Corporate Centers. Im Umgang mit diesen Gruppen, die ebenso wie das Usability Engineering ihren Teil zur Benutzerzentrierung bei UBS Schweiz beitragen, ist vermehrt darauf zu achten, die gemeinsamen Interessen, aber auch die Unterschiede in der jeweiligen Herangehensweise herauszustellen und zu kommunizieren, um eine möglichst fruchtbare Zusammenarbeit unter gegenseitiger Wertschätzung zu erreichen (vgl. hierzu auch Siegel und Dray 2001).

Die Mitarbeiterinnen des Usability Engineerings besitzen spezielle Kompetenzen hinsichtlich der Umsetzung von Benutzerinteressen in Form von gebrauchstauglichen Software-Produkten. Es erscheint daher sinnvoll, Benutzerzentrierung im Rahmen der Software-Entwicklung als Thema des Usability Engineerings zu platzieren und zu kommunizieren. Dieser Anspruch kann dadurch unterstrichen werden, dass eine Infrastruktur geschaffen wird, die das Usability Engineering zur Anlaufstelle macht für die Sammlung und Bereitstellung von praktisch anwendbarem, gut fundiertem Usabilitywissen. Rein organisatorisch wäre im Falle von UBS Schweiz jedoch auch das Corporate Center für die Koordination diesbezüglicher Aufgaben geeignet.

7 Massnahmen zur Erhöhung der Benutzerzentriertheit von UBS Schweiz

Im Prinzip können sämtliche organisatorischen und methodischen Vorgaben, wie beispielsweise eine Entwicklungsrichtlinie oder ein Methodenhandbuch, als Infrastruktur aufgefasst werden. In diesem Sinne wären nahezu alle im folgenden vorgeschlagenen Massnahmen Infrastrukturmassnahmen. Es gibt jedoch ganze Forschungsweige, die sich mit elektronischen Werkzeugen zur Anwendung von Usabilitywissen und zur Förderung benutzerzentrierter Aktivitäten beschäftigen, die gesondert betrachtet werden sollen. Das vorliegende Kapitel differenziert daher die mehrheitlich strategischen Massnahmen und diese technologischen Infrastrukturmassnahmen, die für UBS Schweiz weitgehend Zukunftsmusik darstellen.

7.1 Strategische Massnahmen

Wie die Analyse des Reifegrads der Benutzerzentrierung bei UBS Schweiz gezeigt hat, ist es dort gelungen, dass der Entwicklungsprozess und die vorhandenen Methoden allmählich auch den Aspekt Benutzerzentrierung beinhalten. Unbefriedigend ist sicherlich noch das Mass der Anwendung, die Erfahrung mit benutzerzentrierten Methoden und der allgemeine Wissensstand bezüglich Benutzerzentrierung. Um dies zu erreichen, muss ein Kulturwandel und damit einhergehende flankierende Massnahmen, wie beispielsweise eine erweiterte Definition von Erfolg in Entwicklungsprojekten, durchgesetzt werden. Dies kann nur durch entsprechenden Buy-in des oberen Managements gelingen.

Potentielle benutzerzentrierte Aktivitäten bei UBS Schweiz liegen im Verantwortungsbereich von Business / BTC (Anforderungsphase) und IT (Design- und Realisierungsphase). Bei allen Massnahmen muss sichergestellt werden, dass die richtige Zielgruppe angesprochen wird.

Die folgende Tabelle zeigt mögliche, geplante oder bereits in Angriff genommene Massnahmen zur Erhöhung der Benutzerzentrierung bei UBS Schweiz. Zu jeder Massnahme wird erläutert, welche Ziele damit verfolgt werden, mit welcher Priorität die Massnahme durchgeführt werden sollte und wie gut die Aussichten auf eine erfolgreiche Umsetzung sind. Die Priorisierung orientiert sich an der Analyse der Reifegrade der verschiedenen Dimensionen und Überlegungen zu Voraussetzungen, die gegeben sein müssen, bevor die jeweilige Massnahme überhaupt erfolgreich umsetzbar ist. Ob die Massnahmen hinsichtlich ihrer Zielsetzung tatsächlich Erfolg bringen, müsste nach der Umsetzung im einzelnen evaluiert werden.

Im Falle der Massnahmen ist die Dimension Ressourcen integraler Bestandteil und letztlich ein entscheidendes Kriterium dafür, ob eine Massnahme überhaupt die Chance hat, realisiert zu werden. Spezifische Massnahmen zur Erhöhung der Ressourcen sind nicht vorgesehen, da Ressourcen an sich keinen Selbstzweck besitzen.

Massnahme	Ziel	Priorität / Erfolgsaussichten
<p>Integration weiterer Aktivitäten in die Entwicklungsrichtlinien (ERL); vorgesehen sind:</p> <ol style="list-style-type: none"> 1. Benutzergruppenbeschreibung; 2. High-Level GUI Spezifikation; 3. Iteratives Prototyping; 4. Detaillierte GUI Spezifikation; 5. Usability Test. 	<p>Förderung der Durchführung früher benutzerzentrierter Aktivitäten; iterative Vorgehensweise; kontinuierliche Benutzereinbeziehung.</p> <p>(→ Dimension Prozessmodell)</p>	<p>Prio 1</p> <p>Die vorgesehenen Aspekte sind bereits weitgehend formuliert und werden vom ERL Team so übernommen. Obwohl die Bemühungen bezüglich des Entwicklungsprozesses zukünftig zugunsten anderer Massnahmen eher zurückgenommen werden sollen, sind diese Massnahmen bedeutend, da ansonsten die Gefahr besteht, dass mit dem nächsten ERL Release bisher Erreichtes wieder verloren geht.</p>
<p>Einführung einer Rolle ‚Usability-Verantwortliche/r‘ in den Entwicklungsrichtlinien</p>	<p>Schaffung einer geschäftsbereichsübergreifenden Rolle, die im gesamten Entwicklungsprozess aktiv wird.</p> <p>(→ Dimension Prozessmodell)</p>	<p>Prio 1</p> <p>In den ERL 4.0 (mit Pilotphase ab Mai 2002) ist die Rolle des ‚Usability Engineers‘ vorgesehen. Sie wird als IT Rolle gehandelt.</p>
<p>Usability Engineering als Fachstelle im Sign-off Prozess</p>	<p>Verstärkung der strategischen Bedeutung des Usability Engineerings und Verdeutlichung der Bedeutung von Usability als erfolgskritische Qualität des Endprodukts.</p> <p>(→ Dimension Prozessmodell)</p>	<p>Prio 3</p> <p>Grundsätzlich wäre diese Funktion zwar sinnvoll. Als Sign-off Fachstelle wäre das Usability Engineering zukünftig jedoch nur noch mit der Bearbeitung von Sign-off Dokumenten beschäftigt und könnte keine weiteren Dienstleistungen mehr anbieten, weshalb dies momentan nicht angestrebt wird.</p>

Tabelle 10: Strategische Massnahmen zur Erhöhung der Benutzerzentriertheit bei UBS Schweiz

Massnahme	Ziel	Priorität / Erfolgsaussichten
Requirementsmanagement aktiv vorantreiben, ggf. Competence Center <i>Requirements Engineering</i> im Einflussbereich oder zusammen mit Usability Engineering.	Integration von Software Engineering und Usability Engineering; Frühe Methoden des Usability Engineerings einbinden (Benutzerprofile, Kontext- und Taskanalyse). (→ Dimensionen Prozessmodell, Methoden / Tools)	Prio 3 Für diesbezügliche Aktivitäten unter Führung des Usability Engineerings fehlen momentan die Kapazitäten. Es ist jedoch empfehlenswert Entwicklungen in diesem Bereich innerhalb UBS Schweiz zu beobachten und ggf. Usabilityaspekte einzubringen. Innerhalb der neuen Organisationseinheit laufen bereits Aktivitäten bezüglich Requirementsmanagement, so dass vermutlich auch Finanzierungsmöglichkeiten bestehen.
Erarbeitung eines gut strukturierten, domänen-spezifischen Styleguides	Integration softwareergonomischer Überlegungen in den Designprozess. (→ Dimensionen Methoden / Tools, Qualifikation der Mitarbeiterinnen)	Prio 1 Befindet sich in Umsetzung. Schwierigkeiten: Kompetenzstreitigkeiten zwischen Usability Engineering, IT und Marketing; unterschiedliche Bedürfnisse und Vorstellungen der verschiedenen Projekte; sehr erfolgsversprechend ist der Ansatz, Entwicklerinnen in den Prozess der Styleguideerstellung einzubinden. Durch die Kooperation zwischen Entwickler- und Usabilityseite ist ein aktiver Schritt hin zu gegenseitiger Akzeptanz gelungen (vgl. hierzu auch Kvavik 1994). Die Aktivitäten werden bisher von allen Seiten positiv aufgenommen. Sowohl IT als auch Business sehen den entsprechenden Bedarf und sind bereit, ihren Anteil zur Finanzierung zu leisten.
Erweiterung des vorhandenen Prototypingtools (Visio) zur erleichterten Anwendung des Styleguides.	Integration softwareergonomischer Überlegungen in den Designprozess. (→ Dimensionen Methoden / Tools, Qualifikation der Mitarbeiter)	Prio 1 Visio wird bereits heute als Prototypingtool verwendet. Eine Anpassung an den neuen Styleguide ist vorgesehen. Dies fördert zum einen wiederum die Zusammenarbeit zwischen Usability Engineering und Entwicklerinnen, andererseits ist es auch ein wichtiger Schritt zur Anwendung des Styleguides. Usability Engineering kann in diesem Zusammenhang die Verbreitung von Visio fördern, indem es das Tool bei den Business / BTC Einheiten bekannt macht.

Tabelle 10: Strategische Massnahmen zur Erhöhung der Benutzerzentriertheit bei UBS Schweiz - Fortsetzung

Massnahme	Ziel	Priorität / Erfolgsaussichten
Gezieltes Consulting, Workshops und Schulungen zu Unterschieden und Gemeinsamkeiten der Herangehensweise von Business / BTC und Usability Engineering und zur Umsetzung von funktionalen und qualitativen Anforderungen in gebrauchstaugliche GUIs.	Verteilung der Verantwortlichkeit für Usability des Endproduktes durch Schaffung eines Problembewusstseins bei Business / BTC. Verhindern, dass zwischen Business / BTC und Usability Engineering Kompetenzstreitigkeiten entstehen. (→ Dimensionen Organisationsstruktur, Kultur / Managementprozesse, Qualifikation der Mitarbeiterinnen)	Prio 3 Gewisser Reflexionsprozess auf Seiten des Usability Engineerings muss zunächst erfolgen, um Unterschiede und Gemeinsamkeiten zwischen den Vorgehen greifbar zu machen. Im Rahmen vom Consulting spricht dann nichts dagegen, dies auch zu thematisieren. Bezüglich Workshops und Schulungen von Business / BTC Leuten muss zunächst der Wert solcher Aktivitäten klar aufgezeigt werden können, um Widerstände wie Ressourcenknappheit und Zeitmangel zu überwinden und die Finanzierung sicherzustellen → Bedarf an weiterer Konkretisierung.
Erarbeiten von Schulungsmodulen bezüglich Design (Anwendung des Styleguides);	Schwachstellen einer zentralen Position und begrenzter Ressourcen überwinden; tatsächliche Wirksamkeit des Styleguides erhöhen. (→ Dimensionen Organisationsstruktur, Qualifikation der Mitarbeiter, Schulungswesen und Wissensmanagement)	Prio 1 Der ehemalige Abteilungsleiter von IT Development Services, der sehr von den Anliegen des Usability Engineerings überzeugt ist, leitet neu die Einheit Schulung. Es ist zu erwarten, dass eine Zusammenarbeit diesbezüglich fruchtbar ist.
Schulungen für Solution Manager ¹⁷ im Sign-off Prozess	Schaffung eines Bewusstseins bezüglich der Bedeutung von Usability als Erfolgsfaktor, vor allem bei Führungskräften innerhalb der IT. (→ Dimension Kultur / Managementprozesse)	Prio 2 Erfolgsaussichten bezüglich Schulungsseite gut (s.o.). Das Interesse auf Seiten der Solution Manager muss jedoch gezielt gefördert werden, was unter anderem durch die Einbeziehung von Usabilityaspekten in die ERL gelingen könnte.

Tabelle 10: Strategische Massnahmen zur Erhöhung der Benutzerzentriertheit bei UBS Schweiz - Fortsetzung

¹⁷ Solution Manager nehmen eine zentrale Rolle im Entwicklungsprozess ein. Sie vermitteln zwischen Business und IT und unterstützen die gesamte Abwicklung des Sign-off Prozesses. Der Solution Manager übernimmt die Verantwortung für den gesamten Prozess und kann entscheiden, welche Aktivitäten relevant sind und welche nicht.

Massnahme	Ziel	Priorität / Erfolgsaussichten
<p>Neuen Usability Champion suchen, am besten hoch angesiedelt auf Ressortebene, so dass auch Durchschlagskraft über Ressortgrenzen hinaus vorhanden ist.</p> <p>Mitarbeit in Gremien; Einflussnahme auf IT Managementboard (ITMM) Vortrag vor GB IT Chef</p>	<p>Förderung von Usability als Erfolgskriterium in GB IT.</p> <p>(→ Dimension Kultur / Managementprozesse)</p>	<p>Prio 1</p> <p>Aufgrund der erst kürzlich erfolgten Reorganisation ist momentan noch unklar, wer diese Rolle einnehmen könnte.</p> <p>→ Dringender Bedarf an diesbezüglicher Recherche.</p>
<p>Sichtbares Feedback zur Usability bankinterner Applikationen geben. Möglichkeiten bieten sich dort, wo online Feedback abgefragt wird oder als Rubrik der Intranetseite des Usability Engineerings (z.B. ‚Die Gurke des Monats‘ verleihen o.ä.).</p> <p>Marketing (Tag der offenen Tür, Beitrag Brown Bag University zum Thema Usability) → gezielt auch Nicht-IT Mitarbeiter ansprechen!</p>	<p>Erhöhung des allgemeinen Usabilitybewusstseins innerhalb der Organisation.</p> <p>(→ Dimension Kultur / Managementprozesse)</p>	<p>Prio 1</p> <p>Derartige bewusstseinschaffende Massnahmen erfordern auf längere Sicht signifikante Ressourcen. Dies bedeutet, dass ein Sponsor gefunden werden muss, der für die Finanzierung solcher Aktivitäten aufkommt. Dies könnte vielleicht der Usability Champion sein.</p>
<p>Diskussionsforen (News-groups) zum Thema ‚Benutzerzentrierung im Software-Entwicklungsprozess‘.</p>	<p>Kommunikation aller Stakeholder im Rahmen von benutzerzentrierten Aktivitäten fördern.</p> <p>(→ Dimensionen Qualifikation der Mitarbeiterinnen, Kultur / Managementprozesse)</p>	<p>Prio 3</p> <p>Momentan ist dieses Thema noch zu wenig präsent. Der Bedarf an einem solchen Diskussionsforum sollte zunächst evaluiert und ggf. geweckt werden. Es ist unklar, ob sich ein Sponsor für ein solches Projekt finden würde.</p>

Tabelle 10: Strategische Massnahmen zur Erhöhung der Benutzerzentriertheit bei UBS Schweiz - Fortsetzung

Massnahme	Ziel	Priorität / Erfolgsaussichten
Schulungen zu frühen Methoden; Methodenhandbuch; Brown-Bag University ¹⁸ zu diesem Thema	Sichtbarmachen des Angebots von Methoden, die früh im Entwicklungsprozess einsetzen. (→ Dimension Qualifikation der Mitarbeiter)	Prio 2 Momentan besteht das Problem, dass noch sehr wenig Erfahrung mit frühen Methoden vorhanden ist. Ein Methodenhandbuch besteht zwar, ist jedoch nicht unbedingt als ‚Kochbuch‘ geeignet, um die frühen Methoden danach anwenden zu können. Die Kosten für die Erstellung und Durchführung eines Schulungsmoduls müssen durch eine entsprechende Nachfrage gedeckt werden. Dies ist momentan fraglich. Ausserdem ist der personelle Aufwand sehr hoch, weshalb Schulung vermutlich nur unter Hinzuziehung externer Fachleute realisierbar ist (verschärft das Finanzierungsproblem).
Benutzerfreundliche Hilfsmittel online anbieten (Methoden und Anleitungen, Checklisten, Styleguides, GUI Guidelines).	Sichtbarmachen des Usability Engineerings als Anlaufstelle für Fragen der Benutzereinbeziehung. (→ Dimensionen Qualifikation der Mitarbeiterinnen, Schulungswesen und Wissensmanagement)	Prio 1 Die Veröffentlichung auf dem Intranet ist kein Problem. Die GUI Guidelines des Usability Engineerings sind bereits online verfügbar. Im Rahmen der ERL Einbindung werden auch weitere Dokumente online verfügbar sein. Das proaktive Sammeln von Feedback zur Optimierung des Hilfsmittelangebots ist aus Ressourcengründen momentan noch nicht etabliert.
Aktiv Feedback zu diesen Hilfsmitteln einholen.		

Tabelle 10: Strategische Massnahmen zur Erhöhung der Benutzerzentriertheit bei UBS Schweiz - Fortsetzung

7.2 Infrastrukturmassnahmen

Für die erfolgreiche Ausübung von benutzerzentrierten Aktivitäten im Rahmen eines Software-Entwicklungsprozesses ist das Vorhandensein von Know-how eine Grundvoraussetzung. Dieses Know-how muss im Rahmen der Einführung von benutzerzentrierten Prozessen **entwickelt** und **implementiert** werden und im Laufe der Zeit **systematisch vergrössert** werden. Wie die Analyse des Ist-Zustands der Benutzerzentrierung bei UBS Schweiz gezeigt hat, ist der Reifegrad der Qualifikation der Mitarbeiter und des Schulungswesens und Wissensmanagements deutlich geringer als der Reifegrad des Prozessmodells und der Methoden- und Toolentwicklung. Für eine breite Durchsetzung von benutzerzentrierten Aktivitäten nicht nur innerhalb einzelner Projekte, sondern organisationsweit, ist es bedeutend, dass in allen relevanten Bereichen das entsprechende Know-how vorhanden ist. Dem Prozess der Know-how Entwicklung und Know-how Verbreitung ist daher auch auf der Ebene der Infrastruktur besondere Aufmerksamkeit zu schenken, wobei jedoch zu betonen ist, dass bei jeder Art von Wissensmanagement technologische Massnahmen allein keine Erfolgsgarantie darstellen. Für das Wissensmanagement selbst scheint vielmehr genau das zu gelten, was in dieser Arbeit auch für die Anwendung von benutzerzentrierten Vorgehen postuliert wird: Es müssen die notwendigen Randbedingungen geschaffen werden, um die Mitarbeiterinnen dazu zu veranlassen, ihr Wissen zu teilen, was durch die Entwicklung von Anreizsystemen gefördert werden kann,

¹⁸ Brown-Bag University ist eine Einrichtung bei UBS Schweiz, bei der über Mittag Vorträge zu bestimmten Themen gehalten werden. Alle interessierten Mitarbeiterinnen können sich anmelden und kostenlos an diesen Veranstaltungen teilnehmen. Durch entsprechende Kommunikation können auch gezielt bestimmte Gruppen von Mitarbeitern angesprochen werden.

die jedoch wiederum nicht ohne eine entsprechende Unternehmenskultur wirksam werden (Bullinger et al. 1998).

Im folgenden sollen einige fortschrittliche, bisher meist in Form von Forschungsprototypen vorliegende Tools vorgestellt werden. Im ersten Abschnitt werden Tools beschrieben, die die Anwendung geeigneter Guidelines, Styles und Patterns unterstützen, um damit die Ergonomie der zu entwickelnden Benutzerschnittstelle zu verbessern. Die Anwendung dieser Tools ist begrenzt auf eine Unterstützung des Designprozesses. Im zweiten Abschnitt geht es um Hilfsmittel, die eingesetzt werden, um die benutzerzentrierten Aktivitäten im Rahmen des gesamten Entwicklungsprozesses zu fördern und das diesbezüglich vorhandene Wissen innerhalb der Organisation zu managen.

7.2.1 Tools zur Unterstützung des Designprozesses

Die Idee, mit Hilfe von ergonomischen Standards die Entwicklung von Benutzerschnittstellen zu optimieren ist in den letzten 15 Jahren sehr populär geworden. Ganz egal, welche benutzerzentrierten Methoden sonst noch zur Anwendung kommen oder wie der Entwicklungsprozess aussieht, man kann davon ausgehen, dass die Entwicklungsprojekte von der Anwendung von Styleguides und Guidelines profitieren (Reed et al. 1999).

Die Ziele solcher Styleguides und Guidelines sind einerseits, die Konsistenz im Look & Feel der Anwendungen zu erhöhen, andererseits kann mit ihnen aber auch usabilityrelevantes Wissen verbreitet werden. Auch bei UBS Schweiz stehen GUI Guidelines und Styleguides zur Unterstützung der Gestaltung von Benutzerschnittstellen zur Verfügung (siehe hierzu auch den vorigen Abschnitt).

Um diese Form des Wissenstransfers jedoch möglichst wirksam einzusetzen, müssen folgende Probleme überwunden werden:

- Guidelines sind in der Regel auf einem sehr hohen Abstraktionsniveau oder aber so spezifisch, dass ihre Anwendung erschwert ist;
- Entweder fehlt den Inhalten der theoretische Hintergrund oder dieser wird nicht kommuniziert, was den Erwerb von Usabilitywissen verhindert;
- Guidelines sind für die Entwickler schwer verständlich bzw. sind schwer auf die konkreten Fragestellungen des Projektes anzupassen;
- Guidelines sind statisch und werden in der Regel selten überarbeitet. Ein Optimierungsprozess auf der Basis der gemachten Erfahrungen fehlt meistens.

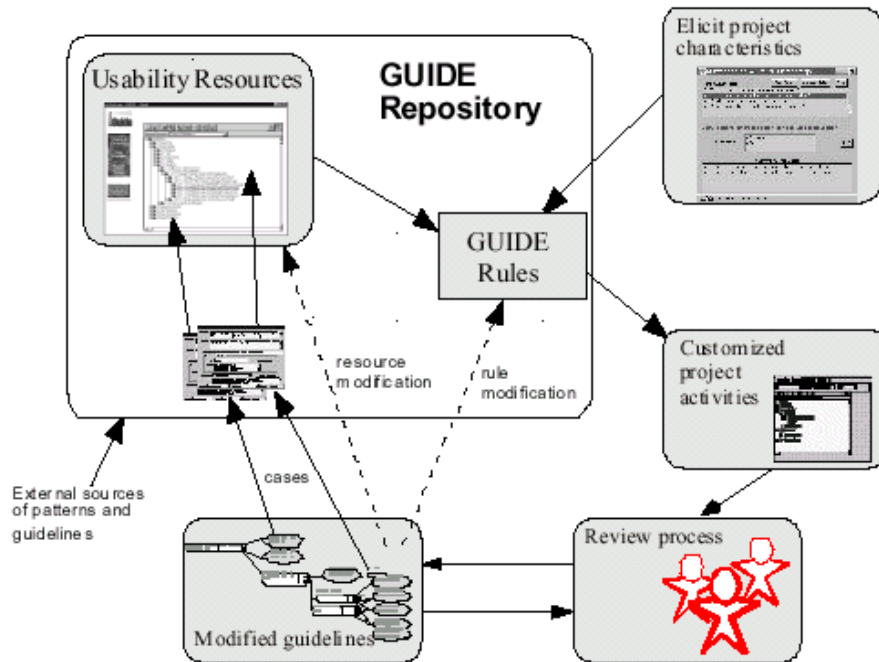
In der Vergangenheit gab es diverse Vorschläge für elektronische Werkzeuge zur Arbeit mit Guidelines bzw. zur Überprüfung der Einhaltung von Guidelines, die diese Mängel überwinden helfen sollten. Z.B. der User Interface Design Assistant «IDA» (Reiterer 1995), das Guideline Management System «Sherlock» (Grammenos et al. 2000), «inra» (Wandke und Hüttner 2000) oder «GUIDE» (Henninger 2001b).

Das Anspruchsniveau der genannten Tools ist sehr unterschiedlich. «inra» bspw. begnügt sich damit, mittels eines Hypertextsystems zu allen Punkten der Guidelines interaktive Beispiele und Szenarien sowie theoretische Hintergründe anzubieten. Experimente haben gezeigt, dass diese Art des Wissensangebots zumindest geeignet ist, um Usabilitywissen erfolgreich anzuwenden und zu transferieren. Zudem scheint die Arbeit mit dem Tool für die Benutzerinnen motivierend zu sein. Allerdings bietet «inra» wenig oder keine Unterstützung darin, die dargestellten Beispiele auf den jeweiligen Problemkontext anzuwenden.

Mit sogenannten «Usability Patterns» wird versucht, Designvorschläge in Abhängigkeit vom jeweiligen Problemkontext anzubieten. Die Patterns repräsentieren dabei eine nachgewiesenermassen brauchbare Lösung für ein vorliegendes Designproblem, das heisst, sie stellen validiertes Wissen dar. Zur Darstellung von Usability Patterns gibt es diverse Vorschläge. In der Regel besteht ein Pattern aus der Beschreibung des Problems, des Kontexts, der Lösung und einem veranschaulichenden Beispiel (van Welie et al. 1999), (Tidwell 1999). Sutcliffe schlägt für die Repräsentation von Patterns die Einbindung in sogenannte «Claims» vor, in denen die Patterns mit Anwendungsszenarien kombiniert werden, was die Diskussion von Vor- und Nachteilen einer Lösung im jeweiligen Problemkontext ermöglicht (vgl. hierzu auch Kap. 3.1.4). Durch das Angebot von zugrundeliegenden Theorien soll zudem Usabilitywissen vermittelt und die jeweilige Lösung wissenschaftlich untermauert werden (Sutcliffe 2000).

Henninger stellt eine Methode vor, um die Anwendung von Usability Guidelines kontextbezogen zu ermöglichen und verfolgt damit im Prinzip die Idee der Usability Patterns. Bei seinem Ansatz wird nicht nur vorhandenes Usabilitywissen angeboten, sondern es wird auch erweitert und neues Wissen wird generiert. Über einen Reviewprozess erhält das System Feedback bezüglich der Qualität und Angemessenheit der vorgeschlagenen Guidelines. Der Reviewprozess kann zu einer Modifikation der Guidelines selbst oder des Auswahlprozesses führen, was einen Lernprozess des Systems und damit der Organisation darstellt.

Der Forschungs-Prototyp «GUIDE» (Guidelines for Usability through Interface Development Experiences) verdeutlicht, wie Softwareentwicklerinnen bei Bedarf mit den passenden Usability Quellen versorgt werden können. Der Ansatz von «GUIDE» ist problemorientiert, d.h. das Problem wird anhand von Attributen charakterisiert und unter Anwendung eines Kontextmodells werden anschliessend regelbasiert geeignete Designvorschläge gemacht. (Henninger 2001b, Henninger et al. 1995)



(Henninger 2001b, S. 4)

Abbildung 17: Die Verwendung und Modifikation von Usability Quellen

Durch den integrierten Feedbackprozess (vgl. Abbildung 17) ist «GUIDE» in der Lage, das vorhandene Wissen unter Berücksichtigung realer Erfahrungen zu aktualisieren. «GUIDE» ist damit vermutlich auch geeignet, den letzten Punkt der eingangs erwähnten Mängelliste bezüglich Guidelines und Styleguides zu überwinden. Allerdings ist zu vermuten, dass gerade durch den hohen Komfort bei der Auswahl geeigneter Designlösungen der gewünschte Wissenstransfer entfällt. Dieser ist jedoch von grösster Bedeutung, denn es darf nicht übersehen werden, dass weder Guidelines noch Styleguides, selbst bei genauester Anwendung, eine Garantie für die Usability des Endproduktes sind.

7.2.2 Tools zur Optimierung des Entwicklungsprozesses

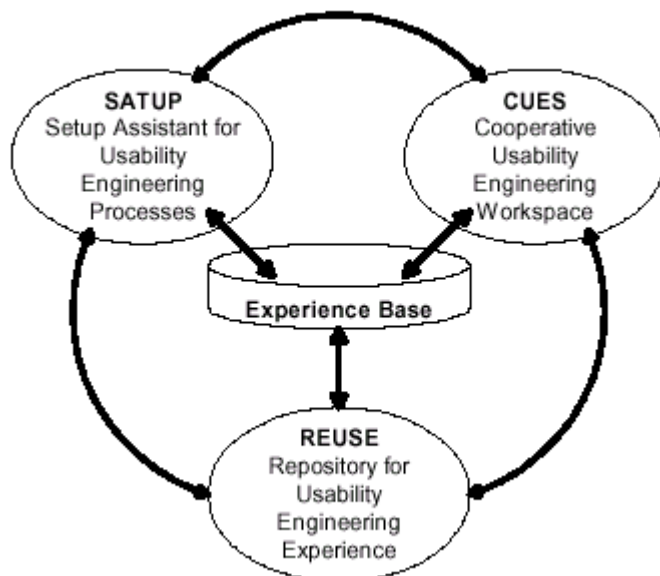
Die selbe Methode, die mit «GUIDE» zur Ermittlung optimaler Guidelines und Patterns verwendet wird, kann auch auf die Optimierung des gesamten Entwicklungsprozesses angewendet werden. Mit «BORE» (Building an Organizational Repository of Experiences) stellt Henninger einen anderen Forschungs-Prototypen vor, der dazu dienen soll, *best practices* innerhalb einer Entwicklungsorganisation zu erfassen und verfügbar zu machen und gleichzeitig den Entwicklungsprozess optimal auf den jeweiligen Projektkontext abzustimmen. (Henninger 2001a, Henninger 1997)

Ganz ähnliche Ideen werden auch mit [Evidence-based Usability Engineering](#) verfolgt. Dabei handelt es sich um einen Usability Engineering Ansatz, der auf dem Nachweis von Erfolg einzelner Aktivitäten und Methoden basiert, mit dem die Optimierung aller benutzerzentrierten Aktivitäten innerhalb der Software-Entwicklung angestrebt wird. Unterstützt wird dieser Ansatz durch eine Plattform namens ProUSE, die hier stellvertretend für alle Tools zur Unterstützung des organisationellen Lernens, näher beschrieben und diskutiert wird.

ProUSE besteht aus einer Wissensbasis und drei logischen Komponenten (vgl. Abbildung 18):

1. Komponente SATUP, die die Zusammenstellung geeigneter Methoden ermöglicht;
2. Komponente CUES, die die Sammlung von Usability Engineering Know-how durch eine gemeinsame Arbeitsumgebung fördert;
3. Komponente REUSE, die konsolidiertes Usabilitywissen (*best practices*) in Form von Anleitungen, Artefakten, Codefragmenten, Templates usw. verfügbar macht.

Evidence-based Usability Engineering optimiert sowohl das Prozessmodell als auch die Durchführung der einzelnen Prozesse. Auf der Basis einer UMM Reifegradanalyse werden geeignete benutzerzentrierte Aktivitäten (*base practices*) in das Prozessmodell der Organisation integriert. Mit Hilfe der ProUSE-Komponente SATUP werden dann die benutzerzentrierten Aktivitäten für den jeweiligen Entwicklungsprozess zusammengestellt. Hierbei kommt ein Kontextmodell zur Anwendung, das Faktoren, wie z.B. Projektart, Dauer und verfügbare Ressourcen berücksichtigt. Die Anwendung der auf diese Weise zusammengestellten benutzerzentrierten Aktivitäten, die in den bestehenden Software-Entwicklungsprozess integriert werden können, wird durch geeignete Anleitungen, Tools und Dokumentvorlagen unterstützt, die in REUSE angeboten werden. Durch die zentrale Sammlung der in jedem Projekt erstellten Dokumente innerhalb von CUES wird die weitere Aufbereitung, Erfahrungssammlung und Weitergabe der Erfahrung an andere Projekte gefördert. Damit wird eine Wissensbasis geschaffen, die einen organisationellen Lernprozess ermöglicht. (Metzker und Reiterer 2002)



(Metzker 2001, S 8)

Abbildung 18: Die Komponenten von ProUSE

ProUSE ist damit geeignet, um folgende Aspekte der Benutzerzentrierung einer Organisation zu fördern:

- Es erlaubt die Integration von benutzerzentrierten Aktivitäten in den Standardentwicklungsprozess;
- Es unterstützt die fortlaufende Optimierung des Prozessmodells;
- Es unterstützt die flexible Instanzierung des generischen Prozessmodells, angepasst an den aktuellen Projektkontext;
- Die Zusammenstellung von *base practices* ist methodenunabhängig, wodurch die Freiheitsgrade für Entwicklerinnen erhalten bleiben;
- Bei Bedarf liefert das Tool Methodenunterstützung und fördert die strukturierte Vorgehensweise bei der Durchführung von *best practices*;
- Die Methoden- und Toolauswahl entspricht der Organisation und wird an das jeweilige Projekt angepasst;
- ProUSE stellt allen Anwenderinnen benutzerzentrierter Methoden grundlegendes Know-how zur Verfügung;
- Die Erstellung von projektrelevanten Dokumenten wird durch das Angebot von Vorlagen und Beispielen unterstützt;
- ProUSE erlaubt die Aufarbeitung von usabilityrelevantem Wissen;
- Individuelles Lernen wird durch das Angebot von theoretischem und praktischem Wissen unterstützt;

- Organisationelles Lernen wird mittels eines Evaluations- und Feedbackprozesses unterstützt. Hierbei werden sowohl der Prozess als auch die einzelnen Methoden optimiert;
- Das Wissen wird in praxisnaher Form angeboten;
- ProUSE ist geeignet, den Aufwand für die Pflege des Entwicklungsprozesses und der methodischen und physischen Infrastruktur sowie für die interne Ausbildung zu reduzieren;
- ProUSE erleichtert die Unterstützung aller relevanten Projekte.

Damit all diese Aspekte tatsächlich effizient unterstützt werden, müssen jedoch folgende Erfolgsfaktoren erfüllt sein:

- ProUSE selbst muss eine hohe Usability aufweisen → bedingt die Anwendung von Usability Engineering Methoden bei der Entwicklung des Tools;
- Innerhalb der Organisation muss ein hoher Implementierungsgrad vorhanden sein, das heisst, alle Projekte müssen bereit sein, ProUSE tatsächlich zu verwenden und zu pflegen → die Verantwortlichen in den Projekten müssen bereits ein gewisses Bewusstsein bezüglich der Usabilityproblematik besitzen, sie müssen die Arbeit mit ProUSE als Erleichterung des Prozesses wahrnehmen und von dessen Effektivität und Effizienzsteigerung überzeugt sein;
- Die Praktikabilität und Kosteneffizienz von ProUSE muss nachweisbar sein → dies bedingt, dass (1) der Wert einzelner benutzerzentrierter Aktivitäten und Methoden tatsächlich evaluiert werden kann, (2) die Auswahl geeigneter Aktivitäten tatsächlich formalisiert werden kann und (3) Erfahrungen mit benutzerzentrierten Aktivitäten in einem Mass angewendet und auch in das System eingespeist werden, dass eine Auswertung überhaupt möglich ist.

Während die ersten beiden Erfolgsfaktoren zumindest bei entsprechender Unterstützung durch das Management vermutlich realisierbar sind, bleibt es noch zu klären, inwieweit der dritte und wichtigste Erfolgsfaktor einzuhalten ist. Grundsätzlich muss davon ausgegangen werden, dass es sehr schwierig ist, den tatsächlichen Nutzen von Usability Engineering Aktivitäten konkret zu beziffern. Im Falle der Anwendung einzelner Methoden bzw. Methodenvarianten dürfte diese Rechnung noch schwieriger sein. Ob es möglich und sinnvoll ist, mit Hilfe eines Tools alle Randbedingungen zu berücksichtigen, die für eine optimale Zusammenstellung von Prozessaktivitäten relevant sind, ist zudem sehr (vgl. hierzu auch die Einschätzung von Prestipino 2002, S. 62f).

7.2.3 Bewertung und Umsetzbarkeit der toolbasierten Ansätze bei UBS Schweiz

Die bereits erwähnten Styleguide-Aktivitäten bei UBS Schweiz könnten dahingehend ausgeweitet werden, dass auf dem Intranet zu den entsprechenden Kapiteln in den Guidelines und Styleguides Beispiele und Code angeboten werden, der zur Nachahmung bzw. Wiederverwendung einlädt. Erste Ansätze hierzu sind bereits angedacht und entsprechen in der Ausprägung ungefähr dem Tool «inra». Erste Erfahrungen mit diesem Vorgehen zeigen, dass die direkte Einbeziehung von Entwicklern automatisch zu Feedback führt, da das Thema Look & Feel von allen Seiten sehr engagiert diskutiert wird. Dennoch könnte es für die Zukunft sinnvoll sein, Feedbackprozesse zu etablieren, die z.B. die systematische Sammlung von Erfahrungen mit der Anwendung des Styleguides erlauben – bisher wird Feedback vor allem im Rahmen von Beratung zur Styleguideanwendung durch Vertreterinnen des Usability Engineerings gesammelt. Die Weiterentwicklung der Styleguides und vor allem der entsprechenden GUI-Elemente im Zuge technologischer Veränderungen (wie bspw. einer neuen Browsergeneration) stellen eine kontinuierliche Pflegeaufgabe dar, für die das entsprechende Budget bereitgestellt werden muss. Dies kann kaum durch einen toolimmanenten Feedbackloop abgedeckt werden.

Generelles Entwicklungspotential besteht bei UBS Schweiz im Bereich früher benutzerzentrierter Methoden. Hier ist das Ziel, nicht nur die Entwickler, sondern vor allem auch die Business- / BTC-Vertreterinnen anzusprechen und entsprechendes Know-how zur Verfügung zu stellen.

Aktivitäten im Bereich der Anforderungsanalyse und –spezifikation sind bei UBS Schweiz heute bereits weit verbreitet. Die angewandten Methoden haben jedoch nicht unbedingt den Fokus auf der Usability des Endproduktes.

Zunächst muss versucht werden, den Business- / BTC-Vertretern die Hilfestellungen bekannt zu machen, die das Usability Engineering im Rahmen der Requirementsphase bereitstellt. Dies bedingt die Erkenntnis, dass die bisherigen Methoden teilweise problematische Auswirkungen auf die Produktentwicklung haben und daher verbesserungswürdig sind. Um dies zu erreichen muss zunächst im Rahmen von Consultingaktivitäten das entsprechende Bewusstsein geweckt werden (siehe hierzu auch die vorgestellten strategischen Massnahmen). Bezüglich des Angebots geeigneter Methoden besteht von seiten Usability Engineering die Einschätzung, dass bestimmte frühe Methoden, wie bspw. Kontext- und Aufgabenanalyse kaum mit Hilfe einer einfachen Anleitung adäquat durchführbar ist. Das seit kurzem vorhandene Methoden-

handbuch dient daher auch eher zu Marketingzwecken. Das Angebot zahlreicher Beispielsituationen und Anwendungen könnte hilfreich sein. Leider sind solche Erfahrungen mit Usability Engineering Methoden noch kaum vorhanden.

In dieser Situation erscheint der Einsatz eines Tools wie ProUSE verfrüht, zumal davon auszugehen ist, dass sich mangels erkanntem Bedarf keine Sponsoren für dessen Entwicklung finden würden. Es könnte jedoch durchaus sinnvoll sein, eine Plattform einzurichten, auf der alle Projekte z.B. ihre Anforderungen sammeln um den *Requirements Engineering* Prozess und die Wiederverwendbarkeit von Anforderungen zu fördern (dies ist bei UBS Schweiz im Rahmen der Bemühungen um ein besseres Requirements Management tatsächlich vorgesehen und müsste vom Usability Engineering entsprechend ausgenutzt werden). Denkbar wäre auch, die Profile von Benutzergruppen zentral abzulegen und Erkenntnisse bezüglich unterschiedlichen Anforderungen zu dokumentieren. Da in sehr vielen Projekten gleiche oder ähnliche Benutzergruppen auftreten (z.B. die verschiedenen Typen an Kundenberaterinnen), sollte es so möglich sein, über kurz oder lang sehr differenzierte Charakterisierungen zu erhalten und bei den Projekten die Wahrnehmung von Unterschieden zwischen den Benutzergruppen zu fördern.

7.3 Zusammenfassung

Initiativen zu einer Verbesserung der Usability von Endprodukten durch vermehrte Benutzerzentrierung werden bei UBS Schweiz bisher ausschliesslich durch das Usability Engineering lanciert. Der vorhandene Reifegrad bezüglich Managementaktivitäten und Kultur bei UBS Schweiz ist noch zu gering, um von einem durchgängigen Bewusstsein hinsichtlich der Usabilityproblematik auszugehen. Die bedeutet zum einen, dass Bemühungen vermehrt auf dieser Ebene angesiedelt werden müssen. Ausserdem wird die begrenzte Kapazität der Usability Engineering Einheit in nächster Zeit noch der limitierende Faktor bei allen Aktivitäten sein.

Wie die dargestellten Erfolgsaussichten zeigen, sind die strategischen Massnahmen zur Steigerung der Benutzerzentrierung bei UBS Schweiz weitgehend realistisch bezüglich ihrer Umsetzung. Es wurden diverse Massnahmen gefunden, die auf der Ebene der Kultur und der Managementaktivitäten ansetzen. Eine höhere Priorisierung dieses Bereichs scheint durchaus sinnvoll, wenngleich sich diese Massnahmen in der Regel schwerer operationalisieren lassen als andere (wie genau findet man beispielsweise einen Usability Champion?). Ideen in diesem Bereich müssen vorderhand selbst entwickelt werden, da das Thema in der Literatur noch stiefmütterlich behandelt wird.

Obwohl andere Massnahmen, z.B. in Bezug auf das Prozessmodell oder im Bereich Methoden und Tools aufgrund der erstellten Analyse weniger prioritär verfolgt werden müssten, ist es offensichtlich, dass zur Erhaltung des Status quo auch in diesen Bereichen kontinuierlich gewisse Anstrengungen notwendig sind.

Ein ständiges Problem wird auch in Zukunft die Ressourcenfrage sein. Diese verschärft sich seit der letzten Reorganisation mit dem Anspruch, 95% aller Aufwendungen auf Projekte zu buchen. Andererseits ist davon auszugehen, dass bei wachsendem Bewusstsein auch die Bereitschaft möglicher Auftraggeber und Sponsorinnen zur Unterstützung von strategischen Aktivitäten des Usability Engineerings steigt.

Der Ausbau der technischen Infrastruktur steckt noch in den Kinderschuhen, ist aber angestossen. Es ist davon auszugehen, dass die Infrastruktur zusammen mit dem Reifegrad der Organisation wächst. Momentane Aktivitäten sollten dazu beitragen die Voraussetzungen für spätere Tools wie ProUSE zu schaffen, indem bereits heute wichtige Informationen zentral gesammelt werden. Es wird hier auch weiterhin der Grundsatz gelten: Keine Tools ohne Nachfrage. Generell ist bei allen toolbezogenen Aktivitäten die enge Einbindung der Entwicklerinnen suchen.

8 Erfahrungen und Ausblick

Die vorliegende Arbeit ist ein Pamphlet. Könnte ich ihr einen anderen Titel geben, würde ich sie „Die benutzerzentrierte Software-Entwicklungsorganisation – Ein Paradigma“ nennen. Obwohl der alte Titel nach wie vor mit dem Inhalt übereinstimmt, gibt er nicht genau die Erkenntnisse wieder, die ich im Laufe der letzten Monate gewonnen habe.

Ausgangspunkt dieser Arbeit war meine Tätigkeit als Usability Consultant bei UBS Schweiz und die Überlegungen, wie man dort den Wirkungsgrad der kleinen Usability Engineering Einheit erhöhen könnte. Ideen hierfür gab es jede Menge. Manche davon waren ganz offensichtlich sinnvoll, wie beispielsweise die Einbindung von Usability Engineering Aktivitäten in die Entwicklungsrichtlinien (ERL). Andere stiessen auf weniger Gegenliebe, wie beispielsweise die Idee, ein Knowledge Management Tool für benutzerzentrierte Vorgehensweisen aufzubauen (ja genau, so was wie ProUSE). Ich hatte die Vision, dass es doch möglich sein müsste, diese Ideen zu strukturieren und argumentativ besser verkaufen zu können.

Die Auseinandersetzung mit Regelwerken und Normen aus dem Bereich der Software-Entwicklung und des Usability Engineerings hat mich einige Wochen gekostet. Was für ein unglaubliches Gewirr mit Anlehnungen, Ausweitungen, Verknüpfungen und feinen kleinen Unterschieden! Am Ende war ich mir sicher (und das ist nicht nur meine Erkenntnis und auch nicht ganz neu), Usability Engineering und Software-Engineering gehören zusammen. Zudem hat mich die Idee des Reifegradmodells überzeugt.

Das Projekt, ein eigenes Reifegradmodell zu erstellen, hat sich als ziemlich gross herausgestellt. Vermutlich ist es doch kein Zufall, dass i.d.R. an einem solchen Konzept sehr viele Leute aus Universität und Praxis über mehrere Jahre hinweg zusammenarbeiten. Entsprechend unvollkommen ist auch das vorliegende Ergebnis. Hier und da sind sicherlich Widersprüchlichkeiten und Lücken vorhanden. Ob es wirklich **diese 7** Dimensionen braucht, ob Qualifikation von Mitarbeiterinnen sowie Schulungswesen und Wissensmanagement nicht doch eine gemeinsame Dimension sind und ob Ressourcen wirklich extra stehen sollen ist natürlich nicht abschliessend geklärt. Besonders hilfreich als Gedankengerüst war für mich die Dimension Managementaktivitäten und Kultur, die von Anfang an im Zentrum meiner Überlegungen gestanden hat. Auch die Organisationsstruktur scheint ein heisses Eisen zu sein. Die Newsgroup zu diesem Thema ist jedoch leider eingeschlafen und vermutlich scheitert diese Dimension schlicht an der Unmöglichkeit, alle Organisationen über einen Kamm zu scheren. Mein Versuch, wenigstens die grundlegenden Gedanken hierzu zu formulieren, ist sicherlich noch nicht ganz befriedigend. Was für mich jedoch heute noch genauso wie am Anfang dieser Arbeit zweifelsfrei feststeht ist, dass es mehr als nur die Dimension des Prozessmodells braucht. «Usability» und «Benutzerzentrierung» sind nicht nur Eigenschaften eines Produktes bzw. eines Prozesses. Sie stellen für mich ein grundlegendes Paradigma dar. Sie sind Antworten auf die Frage, weshalb heute in den allermeisten Fällen Software entwickelt wird (nämlich für Benutzerinnen). Vermutlich würde dem niemand laut widersprechen und dennoch behaupte ich, dass für die wenigsten Leute, die mit Software-Entwicklung zu tun haben, diese beiden Begriffe eine Rolle spielen. Damit sich dies ändert, müssen sich die Rahmenbedingungen ändern, unter denen heutzutage Software entwickelt wird. Es wäre naiv zu glauben, dass sich an den Kosten- und Zeitvorgaben etwas ändern lässt, abgesehen davon, dass dies vermutlich auch nicht zu gebrauchstauglicheren Produkten führen würde. Aber an den Erfolgskriterien sollte sich etwas ändern lassen. Ich bin überzeugt davon, dass in dem Moment, in dem die (seriös ermittelte) Usability des Endprodukts Teil der Erfolgsbewertung eines Projektes wird, sich die Situation für alle benutzerzentrierten Aktivitäten dramatisch ändert. Ebenso bin ich davon überzeugt, dass ohne diese Neudefinition von Erfolg jegliche Bemühungen in diesem Bereich letztlich zum Scheitern verurteilt sind.

Die Ansätze von ISO 15504 (SPICE) bezüglich der Bewertung einzelner konkreter Prozesse und eben nicht eines theoretischen Prozessmodells und vor allem die Idee, Aktivitäten nicht nur bezüglich ihrer Existenz, sondern auch bezüglich ihrer adäquaten Durchführung zu bewerten, erscheinen mir nach wie vor sehr interessant. Sie sollten nach Möglichkeit auch in ein Konzept, das auf der Ebene der Organisation ansetzt, integriert werden. Ein Problem dabei ist allerdings, wie man die adäquate Durchführung von benutzerzentrierten Aktivitäten (strategischer Art oder bezogen auf den Entwicklungsprozess) bewerten kann. Diese Frage hängt eng mit der Operationalisierung der verschiedenen Anforderungen an benutzerzentrierte Organisationen zusammen, die in der vorliegenden Arbeit nur angedacht, jedoch nicht konkret ausformuliert ist. Bisher sind die Anforderungen in erster Linie begründet und es werden Beispiele genannt,

was jeweils in diesem Zusammenhang berücksichtigt werden müsste. An welchen Parametern sich jedoch die adäquate Erfüllung der Anforderung messen liesse, bleibt unbeantwortet.

Die Festlegung von Reifestufen im Rahmen eines Reifegradmodells ist ein weiteres ungelöstes Problem. Grundsätzlich erscheint dies sehr sinnvoll. Die Idee, hierüber Anhaltspunkte für die nächsten strategischen Schritte auf dem Weg zu einer benutzerzentrierten Organisation aufgezeigt zu bekommen, ist bestechend. Es hat sich jedoch gezeigt, dass es sehr schwierig ist zu entscheiden, welches Kriterium welcher Reifestufe entspricht. Zudem ist davon auszugehen, dass sich dies für jede Organisation wieder anders darstellt. Auf jeden Fall braucht es diesbezüglich noch sehr viel mehr Erfahrungen und Diskussionen.

Unsicher bin ich, ob die Abstrahierung von der Situation bei UBS Schweiz gelungen ist. Sehr viele Ideen – letztlich auch für das Modell der benutzerzentrierten Organisation – sind das Resultat meiner konkreten Überlegungen im Kontext der UBS. Ob die Ausführungen jeder Art von Entwicklungsorganisation gerecht werden (also auch solchen Organisationen, die im Auftrag entwickeln oder Standardsoftware herstellen), sei dahingestellt.

Die Idee von Carlshamre und Rantzer, zwischen Tiefe und Breite bei der Anwendung von Usability Engineering Prozessen zu unterscheiden, hat meinem Denken einen neuen Fokus gegeben. Die Zahlen von Ericsson, wo trotz eines gut eingeführten Usability Engineering Prozesses in weniger als 2 % der Projekte dieser Prozess tatsächlich durchgeführt wird, waren für mich ernüchternd. Es ist nicht davon auszugehen, dass dies in irgendeiner europäischen Software-Entwicklungsorganisation besser aussieht, Zahlen hierzu konnte ich aber leider nicht finden. Mir wurde klar, dass die Anwendungsbreite das eigentliche Problem ist und die Durchführung eines einzelnen vollständigen Usability Engineering Prozesses längst nicht das Ziel ist (auch wenn es natürlich schön wäre, einmal die Gelegenheit dazu zu erhalten). Grobe Schätzungen über die Situation bei UBS Schweiz zeigen, dass das Usability Engineering von einem erstaunlich hohen Prozentsatz der Projekte immerhin konsultiert wird (über 10 %) und in ca. 5 % aller Entwicklungen von Benutzerschnittstellen ein Usability Test durchgeführt wird. Diese Zahlen sollen und können jedoch nicht darüber hinwegtäuschen, dass in keinem Projekt bei UBS Schweiz ein vollständiger Usability Engineering Prozess durchgeführt wird.

Von Carlshamre und Rantzer stammt auch die ebenso naheliegende Idee, über das *Requirements Engineering* die Welten des Software- und Usability Engineerings zu verknüpfen. Dass dies jedoch auch nicht ganz einfach ist, ist mir in der Vorlesung bewusst geworden, die ich zu diesem Thema an der Universität Zürich besucht habe. Trotz der unverkennbaren Gemeinsamkeiten zwischen herkömmlichen Zielen und Vorgehensweisen des Requirements Managements mit denjenigen des Usability Engineerings in der Anforderungsphase, besteht auch hier ein nicht zu vernachlässigender Kulturunterschied. Auf jeden Fall scheint es mir angebracht, alle Aktivitäten bezüglich Requirements Management bei UBS Schweiz mit Argusaugen zu verfolgen, um auch die Ideen des Usability Engineerings dort einzubringen.

Gerne hätte ich noch mehr zu den Bemühungen bei Ericsson erfahren, aber eine entsprechende Anfrage bei Carlshamre und Rantzer blieb unbeantwortet. Genauso erfolglos war meine Anfrage bei Stephanie Rosenbaum zu ihren gross angelegten Studien bei Hunderten von Usability Fachleuten zu deren strategisch bedeutsamsten Vorgehen und Methoden. Mehr Austausch zu solch einem komplexen Thema wäre notwendig, zumal gerade die Frage nach der Effizienz von Methoden und Vorgehensweisen nach wie vor unbeantwortet ist.

Während der letzten Monate hatte ich die Chance, mich einerseits auf die konzeptionellen Fragen dieser Arbeit zu konzentrieren und gleichzeitig die konkreten, alltäglichen Fragestellungen des Usability Engineerings zu erleben. Ich denke, beide Aspekte sind wichtig! Ich hoffe, dass es mir gelungen ist, die beiden Ebenen fruchtbar zusammenzubringen, und mit dieser Arbeit ein paar konstruktive Ansätze zum Thema ‚Benutzerzentrierung in Software-Entwicklungsprozessen‘ beizusteuern.

9 Glossar

Base Practice	Software Engineering - oder Managementaktivität, deren Ausübung im Hinblick auf den Zweck eines Prozesses notwendig ist. Definiert z.B., was innerhalb eines benutzerzentrierten Prozesses zu tun ist.
Benutzerzentrierter Entwicklungsprozess	Entwicklungsprozess, bei dem die Einbeziehung von Benutzerinnen zur Erhebung und Validierung von funktionalen und qualitativen Anforderungen, zur Evaluation von Designoptionen und zur Abnahme des Endproduktes vorgesehen ist.
Best Practice	Konkrete Anleitung zur Durchführung bestimmter Aktivitäten (<i>base practices</i>) im Rahmen eines Entwicklungsprozesses. Die best practices definieren, wie bestimmte Aktivitäten im Rahmen eines Entwicklungsprozesses durchgeführt werden sollen. Idealerweise werden best practices kontinuierlich evaluiert und optimiert (vgl. hierzu auch Evidence-Based Usability Engineering).
(Process) Capability	Die Fähigkeit eines Prozesses, bestimmte gesteckte Ziele zu erreichen (z.B. Einhaltung des Kosten- und Zeitrahmens, aber auch jede andere Art definierter Qualitätsfaktoren).
Claimanalyse	Analytische Methode zur Ermittlung der im Rahmen eines Szenarios auftauchenden Designelemente (Features) mit ihren positiven und negativen Auswirkungen auf die Usability eines Systems.
Classic GUI	Eine Benutzerschnittstelle, die nach den Vorgaben der GB IT Webplattform bei UBS Schweiz nur unter Verwendung von HTML und wenig JavaScript entwickelt wurde, so dass praktisch Browserunabhängigkeit vorliegt (im Gegensatz zu einem sogenannten Rich GUI).
(Innovations-)Diffusion	Prozess, über den eine Innovation im Laufe der Zeit mit Hilfe bestimmter Kanäle über die Mitglieder eines sozialen Systems oder einer Organisation ausgebreitet wird.
Evidence-based Usability Engineering	Vorgehensweise, bei der die benutzerzentrierten Aktivitäten innerhalb einer Organisation optimiert werden, indem Erfahrungen bei der Durchführung bestimmter Aktivitäten und Methoden ausgewertet und zur Anpassung des Prozessmodells und der best practices verwendet werden.
Feature	Hier verwendet für die (meist funktionale) Eigenschaft bzw. Designelemente einer Software oder eines Systems.
Generic Practice	Implementations- oder Institutionalisierungsaktivität zur Verbesserung der Capability eines Prozesses.
(Process) Maturity	Das Ausmass, in dem ein spezifischer Prozess explizit definiert, organisiert, gemessen, kontrolliert und effektiv ist.
(Process) Performance	Ein Mass der tatsächlichen Resultate, die durch die Anwendung eines Prozesses erzielt werden.

Requirements Engineering (Anforderungstechnik)	1. Das <i>systematische, disziplinierte</i> und <i>quantitativ erfassbare</i> Vorgehen beim Spezifizieren, d.h. Erfassen, Beschreiben und Prüfen von Anforderungen an Software. 2. Verstehen und Beschreiben, was Kunden wünschen oder brauchen. (Glinz WS 2001/2002, S. 1 - 4)
Rich GUI	Eine Benutzerschnittstelle, die unter Verwendung aller möglichen Webtechnologien entwickelt wurde, die es ermöglichen, das Look & Feel einer Desktop-Applikation zu imitieren. Diese Benutzerschnittstellen erfordern zum einen teilweise den Download bestimmter Plug-Ins, andererseits besteht das Risiko, dass sie für Benutzerinnen ausserhalb UBS Schweiz aufgrund des verwendeten Browsertyps oder der Browsereinstellungen nicht mehr funktional sind (im Gegensatz zu einem sogenannten Classic GUI).
Trade-off	(engl.) Aushandeln, Austausch. Hier verwendet in dem Sinne, dass es für einzelne Lösungsoptionen eines Designproblems immer unterschiedliche Argumente gibt, die jeweils die positiven und negativen Auswirkungen einer Lösungsoption behandeln.

10 Literaturverzeichnis

Anderson, R., Organizational Limits to HCI: Conversations with Don Norman and Janice Rohn. *interactions*, 7 (3), 2000. 36-60.

Balzert, H., *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 1998.

Beyer, H. and Holtzblatt, K. (eds.). *Contextual Design. Defining Customer-Centered Systems*. Morgan Kaufmann Publishers, San Francisco, 1998.

Billingsley, P.A., Business: Starting from Scratch: Building a Usability Program at Union Pacific Railroad. *interactions*, 1 (5), 1995. 27-30.

Bloomer, S. and Wolfe, S., The Hiser Group: Pioneering Usability and User Interface Design in Australia. In: *Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground*, (1996), 141-142.

Borgholm, T. and Halskov Madsen, K., Cooperative Usability Practices. *Communications of the ACM*, 42 (5), 1999. 91-97.

Bullinger, H.-J., Warschat, J., Prieto, J. and Wörner, K., Wissensmanagement - Anspruch und Wirklichkeit: Ergebnisse einer Unternehmensstudie in Deutschland. *Information Management* (1), 1998. 7-23.

Carlshamre, P. and Rantzer, M., Business: Dissemination of Usability. Failure of a Success Story. *interactions*, 8 (1), 2001. 31-41.

Carter, J., Incorporating standards and guidelines in an approach that balances usability concerns for developers and end users. *Interacting with Computers*, 12, 1999. 179-206.

CMMISM for Systems Engineering/Software Engineering, Version 1.02. Carnegie Mellon Software Engineering Institute, 2000.

Constantine, L.L., Process Agility and Software Usability: Toward Lightweight Usage-Centered Design, 2001. <http://www.forUse.com/Files/Papers/agiledesign.pdf>

Constantine, L.L. and Lockwood, L.A.D., *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley, 1999.

Constantine, L.L. and Lockwood, L.A.D., Structure and Style in Use Cases for User Interface Design, 2000. <http://www.forUse.com/Files/Papers/structurestyle2.pdf>

Cooper, A., *The Inmates Are Running the Asylum*. SAMS, Indianapolis, Indiana, 1999.

Deutsche Akkreditierungsstelle Technik e.V., DATech-Prüfbaustein Usability-Engineering-Prozess. Leitfaden für die Evaluierung des Usability-Engineering-Prozesses bei der Herstellung und Pflege von Produkten auf der Grundlage von DIN EN ISO 13407. 2001a.

http://www.datech.de/download/deutsch/13-2001_DATech_Baustein_UE-Prozess_1.1.pdf

Deutsche Akkreditierungsstelle Technik e.V., DATech-Prüfhandbuch Gebrauchstauglichkeit. Leitfaden für die software-ergonomische Evaluierung von Software auf Grundlage von DIN EN ISO 9241, Teile 10 und 11. 2001b. http://www.datech.de/download/deutsch/12-2001_handbuch_gebrauchstauglichkeit_V3.1.pdf

Dieli, M., Dye, K., McClintock, M. and Simpson, M., The Microsoft Corporation Usability Group. In: Wiklund, M.E. ed. *Usability in Practice. How Companies Develop User-Friendly Products*, AP Professional, Cambridge, MA, 1994, 327-358.

DIN EN ISO 9000-3: Normen zum Qualitätsmanagement und zur Qualitätssicherung/QM-Darlegung - Teil 3: Leitfaden für die Anwendung von ISO 9001:1994 auf Entwicklung, Lieferung, Installation und Wartung von Computer-Software (1998).

DIN EN ISO 9241: 10 -17: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten

Teil 10: Grundsätze der Dialoggestaltung

Teil 11: Anforderung an die Gebrauchstauglichkeit - Leitsätze

Teil 12: Informationsdarstellung

Teil 13: Benutzerführung

Teil 14: Dialogführung mittels Menü

Teil 15: Dialogführung mittels Kommandosprache

Teil 16: Dialogführung mittels direkter Manipulation

Teil 17: Dialogführung mittels Bildschirmformularen (1996-1999).

DIN EN ISO 13407: Benutzer-orientierte Gestaltung interaktiver Systeme (1999).

Donahue, G.M., Weinschenk, S. and Nowicki, J., Usability Is Good Business, 1999.

<http://www.compuware.com>

DSDM-Consortium, DSDM Manual (Synopsis).

http://www.dsdm.org/products/dsdm_manual/index.asp

Earthy, J., Usability Maturity Model: Human Centredness Scale. Lloyd's Register, project IE2016 INUSE, Deliverable D5.1.4(s), 1998.

Earthy, J., Usability Maturity Model: Processes (TRUMP Version). Lloyd's Register, project IE2016 INUSE, Deliverable D5.1.4(p), 1999.

Erleben, K. and Gebauer, A., EnjoySAP-Erfolgsfaktoren für die Einführung eines benutzerorientierten Software-Entwicklungsprozesses. *HMD*, 36 (212), 2000. 41-56.

Faulkner, X. and Culwin, F., Enter the Usability Engineer: Integrating HCI and Software Engineering. In: *Proceedings of the 5th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education*, (2000), 61-64.

Ferré, X., Incorporating Usability into an Object Oriented Development Process. *Accepted Position Paper for Workshop at INTERACT*, 2001.

Garrison, G., Heath, R. and Jaynes, A., The Usability Group at Reuters: Virtually Global. In: *Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground*, (1996), 137-138.

Glinz, M. Spezifikation und Entwurf von Software. Skript zur gleichnamigen Vorlesung, Universität Zürich, WS 2001/2002.

Gould, J.D., Boies, S.J. and Lewis, C., Making Usable, Useful, Productivity-Enhancing Computer Applications. *Communications of the ACM*, 34 (1), 1991. 75-85.

Grammenos, D., Akoumianakis, D. and Stephanidis, C., Integrated support for working with guidelines: the Sherlock guideline management system. *Interacting with Computers*, 12, 2000. 281-311.

Grudin, J., The Organizational Contexts of Development and Use. *ACM Computing Surveys*, 28 (1), 1996. 169-171.

Henninger, S., An Environment Supporting Software Process Standardization and Innovation, 2001a.

<http://cse-ferg41.unl.edu/bore.html>

- Henninger, S., An Organizational Learning Method for Applying Usability Guidelines and Patterns. In: *Proceedings of the 8th IFIP Working Conference on Engineering for Human-Computer Interaction*, (Toronto, CA, 2001b).
- Henninger, S., Tools Supporting the Creation and Evolution of Software Development Knowledge. In: *Proceedings of the 12th International Conference on Automated Software Engineering*, (Lake Tahoe, NV, 1997), 46-53.
- Henninger, S., Haynes, K. and Reith, M.W., A Framework For Developing Experience-Based Usability Guidelines. In: *Proceedings of the Symposium on Designing Interactive Systems*, (Ann Arbor, MI, 1995), 43-53.
- Hudson, W., Toward Unified Models in User-Centered and Object-Oriented Design. In: Harmelen, M.v. ed. *Object Modeling and User Interface Design*, Addison-Wesley, 2001.
- IBM, Getting Started, 1999. http://www-3.ibm.com/ibm/easy/eou_ext.nsf/
- ISO 9001: Qualitätsmanagementsysteme - Anforderungen (1994).
- ISO 9241-2: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 2: Leitsätze zur Aufgabengestaltung (1992).
- ISO/IEC 9126: Software-Engineering - Qualität von Software-Produkten - Teil 1: Qualitätsmodell (2001).
- ISO/IEC 12119: Informationstechnik - Software-Erzeugnisse - Qualitätsanforderungen und Prüfung (1994).
- ISO/IEC 12207: Informationstechnik - Prozesse im Lebenszyklus von Software (1995a).
- ISO/IEC SPICE Software Process Assessment - Part 1: Concepts and introductory guide (1995b).
- ISO/IEC SPICE Software Process Assessment - Part 2: A model for process management (1995c).
- ISO/IEC SPICE Software Process Assessment - Part 7: Guide for use in process improvement (1995d).
- Jacobson, I., The Use-Case Construct in Object-Oriented Software Engineering. In: Carroll, J.M. ed. *Scenario-based design*, John Wiley & Sons, Inc., New York, 1995.
- Kaderbhai, T., Overcoming Inertia within a Large Organization. How to Overcome Resistance to Usability. In: Trenner, L. and Bawa, J. eds. *The Politics of Usability. A Practical Guide to Designing Usable Systems in Industry*, Springer-Verlag, Berlin, Heidelberg, New York, 1998, 35-48.
- Karat, C.-M., A Business Case Approach to Usability Cost Justification. In: Bias, R.G. and Mayhew, D.J. eds. *Cost-Justifying Usability*, Morgan Kaufmann, San Francisco, CA, 1994, 45-70.
- Kvavik, K.H., Starting and Expanding User-Centered Processes. *interactions*, 1 (7), 1994. 68-70.
- Landauer, T.K., *The Trouble with Computers. Usefulness, Usability, and Productivity*. The MIT Press, Cambridge, Massachusetts, 1995.
- Lederer, A.L. and Prasad, J., Nine Management Guidelines for Better Cost Estimating. *Communications of the ACM*, 35 (2), 1992. 51-59.
- Leffingwell, D. and Widrig, D., *Managing Software Requirements. A Unified Approach*. Addison-Wesley, Reading (MA), 2000.
- Mayhew, D.J. (ed.), *The Usability Engineering Lifecycle. A Practitioner's Handbook for User Interface Design*. Morgan Kaufmann, San Francisco, CA, 1999.
- Mayhew, D.J. and Bias, R.G., Organizational Inhibitors and Facilitators. In: Bias, R.G. and Mayhew, D.J. eds. *Cost-Justifying Usability*, Morgan Kaufmann, San Diego, 1994, 287-318.

- Metzker, E. Evidence-Based Usability Engineering: Seven Thesis on the Integration, Establishment and Continuous Improvement of Human-Centred Design Methods in Software Development Processes *INTERACT, Workshop: Usability throughout the entire development lifecycle. Accepted position paper*, 2001.
- Metzker, E., Fetzter, S. and Weber, M. REUSE: Supporting the Introduction, Establishment and Continuous Improvement of Human Centred Design Processes *Submitted to International Conference on Human Factors in Computing Systems, CHI*, 2001.
- Metzker, E. and Reiterer, H. Use and Reuse of HCI Knowledge in the Software Development Lifecycle: Existing Approaches and what Developers Think. *Submitted to IFIP World Congress*, 2002.
- Nielsen, J., Alertbox: Field Studies Done Right - Fast and Observational, 2002.
<http://www.useit.com/alertbox/20020120.html>
- Nielsen, J., Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier, 1994a.
http://useit.com/papers/guerrilla_hci.html
- Nielsen, J., *Usability Engineering*. Academic Press, New York, 1993.
- Nielsen, J., *Usability Inspection Methods*. John Wiley, New York, 1994b.
- Norman, D.A., Cognitive engineering. In: Norman, D.A. and Draper, S.D. eds. *User Centered System Design*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986, 31-61.
- Norman, D.A., *The Invisible Computer*. MIT Press paperback edition, 1998.
- Paulk, M., Chrissis, M.B., Weber, C. and Perdue, J. The Capability Maturity Model for Software, Version 2B, Pittsburgh, PA 15213-3890, 1997, 92.
- Pelka, A.I., Integration von Software- und Usability-Engineering am Beispiel des V-Modells. Diplomarbeit, Rheinische Friedrich-Wilhelms-Universität, 2000.
- Prestipino, M., Reifegradmodelle zur wissensorientierten Prozessverbesserung bei der Entwicklung gebrauchstauglicher Software. Masterarbeit, Universität Konstanz, 2002.
- Probasco, L., The Ten Essentials of RUP, 2000. http://www.therationaledge.com/content/dec_00/
- Reed, P., Holdaway, K., Isensee, S., Buie, E., Fox, J., Williams, J. and Lund, A., User interface guidelines and standards: progress , issues, and prospects. *Interacting with Computers*, 12, 1999. 119-142.
- Reiterer, H., IDA - A Design Environment for Ergonomic User Interfaces. In: Nordby K., Helmersen P., Gilmore D. and Arnesen S. eds. *Human-Computer Interaction, Interact '95, IFIP Conference*, Chapman & Hall, London, 1995, 305-310.
- Rogers, E.M., *Diffusions of Innovations*. The Free Press, New York, 1983.
- Rosenbaum, S., Humburg, J. and Rohn, J.A., Unpacking Strategic Usability: Corporate Strategy and Usability Research. In: *Proceedings of the Conference on CHI '98 Summary: Human Factors in Computing Systems*, (1998), 205.
- Rosson, M.B. and Carroll, J.M., *Usability Engineering. Scenario-Based Development of Human-Computer Interaction*. Academic Press, San Diego, CA, 2002.
- Saunders, J. and Arnfeld, A., Making Usability Part of the Culture. In: Trenner, L. and Bawa, J. eds. *The Politics of Usability. A Practical Guide to Designing Usable Systems in Industry*, Springer-Verlag, Berlin, Heidelberg, New York, 1998, 69-82.

- Schaffer, E., The Institutionalization of Usability. Human Factors International White paper, 2001.
www.humanfactors.com
- Schönwälder, J. Informatik C, Vorlesungsskript WS 2001/2002, Universität Osnabrück, 2002.
- Schuler, D. and Namioka, A. (eds.). *Participatory Design: Principles and Practices*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1993.
- Siegel, D.A. and Dray, S.M., Business: New Kid on the Block: Marketing Organizations and Interaction Design. *interactions*, 8 (2), 2001. 19-24.
- Sprenger, R.K., *Mythos Motivation. Wege aus einer Sackgasse*. Campus Verlag, Frankfurt / New York, 1998.
- Stelzer, D., Stand des Qualitätsmanagements in der Softwareentwicklung. In: Hummeltenberg, W. ed. *Information Management for Business and Competitive Intelligence and Excellence.*, Wiesbaden, 2000, 313-326.
- Sutcliffe, A., On the Effective Use and Reuse of HCI Knowledge. *ACM Transactions on Computer-Human Interaction*, 7 (2), 2000. 197-221.
- Tidwell, J., Common Ground: A Pattern Language for Human-Computer Interface Design, 1999.
<http://www.mit.edu/~jtidwell/toc.html>
- UBS AG. GB IT. Die Rollen, 2001.
- UBS AG. Jahresbericht 2001, Zürich, 2002.
- UBS AG. Projektmanagement & ERL Distilled 1.0, Internes Dokument, 1999.
- UBS IT Management. Implementation of the IT Strategy of UBS Switzerland: Masterplans 2002 - 2004, UBS Intranet, 2002.
- van Welie, M., van der Veer, G.C. and Eliëns, A., Patterns as Tools for User Interface Design. In: Vanderdonck, J. and Farenc, C. eds. *Tools For Working with Guidelines*, Springer-Verlag, London, New York, 1999.
- Vredenburg, K., Increasing Ease of Use. *Communications of the ACM*, 42 (5), 1999. 67-71.
- Wandke, H. and Hüttner, J., Completing Human Factor Guidelines by Interactive Examples. In: Vanderdonck, J. and Farenc, C. eds. *Tools for Working with Guidelines*, Springer, 2000.
- Wiebe, E.N., Four factors that shape the corporate role of usability. Deep Realities: The fit of usability in business. *ACM Journal of Computer Documentation*, 24 (4), 2000. 220-226.
- Wixon, D.R. and Comstock, E.M., Evolution of Usability at Digital Equipment Corporation. In: Wiklund, M.E. ed. *Usability in Practice. How Companies Develop User-Friendly Products*, AP Professional, Cambridge, MA, 1994, 147-194.
- WM-data AB & Ericsson Radio Systems AB, The Deltamethod Handbook. Manual, 2000.
<http://www.deltamethod.net>
- Wood, J. and Silver, D., *Joint Application Development*. John Wiley & Sons, New York, 1995.
- Wynekoop, J.L. and Senn, J.A., CASE Implementation: The Importance of Multiple Perspectives. In: *Proceedings of the 1992 ACM SIGCPR Conference on Computer Personnel Research*, (1992), 63-74.

11 Anhang

11.1 Vergleich der verschiedenen Usability Engineering Prozessmodelle

Usability Engineering Lifecycle (Mayhew 1999)	Delta (WM-data AB & Ericsson Radio Systems AB 2000)	Contextual Design (Beyer und Holtzblatt 1998)	Scenario Based Design (Rosson und Carroll 2002)	Usage-Centered Design (Constantine und Lockwood 1999)
Wird als gegeben betrachtet!	System Definition (Systemdefinition)	Setting Project Focus (Definition des Projektrahmens und der Zielgruppe)	Root concept: vision, rationale, assumptions, stakeholders (Systemrahmen im Prinzip gegeben; Erarbeitung eines gemeinsamen Verständnisses des Grundkonzepts des Systems)	Als Fragestellung in der Erhebung der Benutzerrollen impliziert. Nicht explizit aufgeführt. Grundsätzlich als gegeben betrachtet
User Profiles (Erstellen von Benutzerprofilen): Benutzerprofile sind Charakterisierungen der verschiedenen Zielgruppen. Jeder Benutzer gehört einer Benutzergruppe an	User Profiling (Erhebung von Benutzerprofilen)	Nicht explizit aufgeführt. Contextual Design verwendet ein Rollenkonzept. Einzelne Benutzer können mehrere Rollen im Zusammenhang mit dem System einnehmen.	Z.T. in Root concept enthalten; als ‚Actors‘ in den Problemszenarien ausformuliert. Vergleichbar mit der Idee der Personas von Alan Cooper (Cooper 1999)	User Role Modeling (Modellierung von Benutzerrollen)
Contextual Task Analysis (Kontextbezogene Aufgabenanalyse)	Task Analysis (Aufgabenanalyse):	Understanding the Customer (Kundendaten zusammentragen)	Analyzing Requirements (Anforderungsanalyse):	Collaborative Requirements Dialog (Anforderungsspezifikation)
Teilweise sind die Aufgaben des Domain Modelings in der Kontexterhebung beinhaltet (Erfassung eines gemeinsamen Vokabulars usw.)	Nicht Bestandteil von Delta	Teilweise sind die Aufgaben des Domain Modelings in der Kontexterhebung beinhaltet (Erfassung eines gemeinsamen Vokabulars usw.)	Nicht Bestandteil von Scenario Based Design	Domain Modeling (Modellierung der Anwendungsdomäne)
Nicht explizit vorhanden! Integriert in Contextual Task Analysis	Ergebnisse der Task Analysis werden in Form von Aktivitätsdiagrammen dokumentiert	Seeing Work (Entwickeln von Arbeitsmodellen)	Implizit in Szenarien enthalten. Keine explizite Verwendung von Modellen.	Task Modeling (Aufgabenmodellierung)

Tabelle 11: Vergleich von 5 bekannten Usability Engineering Prozessmodellen

Usability Engineering Lifecycle (Mayhew 1999)	Delta (WM-data AB & Ericsson Radio Systems AB 2000)	Contextual Design (Beyer und Holtzblatt 1998)	Scenario Based Design (Rosson und Carroll 2002)	Usage-Centered Design (Constantine und Lockwood 1999)
Nicht explizit vorhanden! Integriert in Contextual Task Analysis und Conceptual Model Design. Konsolidierung ist implizites Ziel von Iterationen.	Design Preparations (Bereitstellung von Designratschlägen und Szenarien)	Seeing across Customers (Konsolidierung der Daten)	Nicht explizit vorhanden!	Nicht explizit vorhanden!
Usability Goals (Setzen von Gebrauchstauglichkeitszielen).	Usability Requirements (Formulierung von Usability Anforderungen) Identisch mit Usability Goals.	Nicht Bestandteil von Contextual Design.	Nicht explizit vorhanden! Indirekt Bestandteil von Usability Spezifikationen im Zusammenhang mit Usability Evaluation.	Indirekt und auf hohem Abstraktionsniveau in Form von Usability Kriterien (allgemeine und benutzerrollen-spezifische) abgehandelt.
Platform Capabilities / Constraints (Erhebung von Platfformeigenschaften)	Nicht behandelt!	Nicht Bestandteil von Contextual Design; explizit im Entwicklungsprozess nach hinten geschoben.	Nicht behandelt!	Wird später in Implementationsaktivitäten behandelt.
General Design Principles (Allgemeine Designprinzipien)	Nicht explizit vorhanden, jedoch impliziter Bestandteil des Prototyping Schrittes.	Nicht Bestandteil von Contextual Design.	Nicht explizit vorhanden, jedoch impliziter Bestandteil des Prototyping Schrittes.	Nicht explizit vorhanden, jedoch impliziter Bestandteil des Implementation Modelings.
Neudefinition von Arbeitsabläufen im Work Reengineering auf Systemebene beschränkt.	In diesem Umfang nicht in Delta vorhanden.	Work Redesign (Neugestaltung der Arbeitsprozesse) Die Zielsetzung des Work Redesigns im Contextual Design geht über die Ziele der anderen Prozesse an diesem Punkt hinaus!	In diesem Umfang nicht im Scenario Based Design vorhanden.	In diesem Umfang nicht im Usage-Centered Design vorhanden.
Work Reengineering (Neugestaltung der Arbeitsprozesse)	Nicht explizit vorhanden, implizit durch das Conceptual Design abgedeckt.	System Work Model (Systemdesign / System Arbeitsmodell)	Activity Design (Aktivitätsdesign)	Im Zusammenspiel zwischen Essential Use Cases und Interface Content Modeling implizit vorhanden.

Tabelle11: Vergleich von 5 bekannten Usability Engineering Prozessmodellen – Fortsetzung.

Usability Engineering Lifecycle (Mayhew 1999)	Delta (WM-data AB & Ericsson Radio Systems AB 2000)	Contextual Design (Beyer und Holtzblatt 1998)	Scenario Based Design (Rosson und Carroll 2002)	Usage-Centered Design (Constantine und Lockwood 1999)
Conceptual Model Design (Entwurf eines Konzepts für das Gesamtsystems)	Conceptual Design (Erstellung des Designkonzepts)	User Environment Design (Abbildung der Daten auf ein User Interface)	Information Design (Informationsdesign) Activity Design + Information Design auf dem obersten Abstraktionsniveau entspricht in etwa dem, was an anderer Stelle unter Conceptual Design oder Conceptual Model Design verstanden wird.	Interface Content Modeling (Modellierung des Inhalts der Benutzerschnittstelle)
Konkrete physische Interaktion zwischen System und Benutzer wird im Rahmen der Screen Design Standards und des Detailed User Interface Designs entwickelt.	Impliziter Bestandteil des Prototypings.	Auf Konzeptebene impliziter Bestandteil des Prototypings.	Interaction Design (Interaktionsdesign)	Impliziter Bestandteil des Implementation Modeling und des dazugehörigen Prototypings.
Conceptual Model Mock-ups (Papierprototypen des Konzepts) Prototyping wird als Design-Hilfsmittel verstanden und nicht als explizite Aktivität.	Prototype Design (Entwicklung eines Prototypen)	Prototyping / Iterating with a Prototype (Iteratives Design durch Prototyping)	Prototyping (Entwicklung von Prototypen)	Implementation Modeling (Implementationsmodellierung) Prototyping wird als Design-Hilfsmittel verstanden und nicht als explizite Aktivität.
Iterative Conceptual Model Evaluation (Iterative Evaluation des K)	Impliziter Bestandteil des Prototypings.	Impliziter Bestandteil des Prototypings.	Impliziter Bestandteil des Prototypings.	Impliziter Bestandteil des Prototypings im Rahmen des Implementation Modelings.
Nicht vorhanden!	Nicht vorhanden!	Project Planning and Strategy (Projektplanung und –Strategie)	Nicht vorhanden!	Releaseplanung auf der Basis der essentiellen Use Cases angesprochen. Keine explizite Aktivität.
Screen Design Standards (Standards für den Entwurf der Bildschirmoberfläche)	Impliziter Bestandteil des Prototypings.	Nicht Bestandteil von Contextual Design.	Impliziter Bestandteil des Prototypings.	Impliziter Bestandteil des Prototypings.

Tabelle11: Vergleich von 5 bekannten Usability Engineering Prozessmodellen – Fortsetzung.

Usability Engineering Lifecycle (Mayhew 1999)	Delta (WM-data AB & Ericsson Radio Systems AB 2000)	Contextual Design (Beyer und Holzblatt 1998)	Scenario Based Design (Rosson und Carroll 2002)	Usage-Centered Design (Constantine und Lockwood 1999)
Screen Design Standards Prototyping (Prototyping der Screen Design Standards)	Impliziter Bestandteil des Prototypings.	Nicht Bestandteil von Contextual Design.	Impliziter Bestandteil des Prototypings.	Impliziter Bestandteil des Prototypings.
Impliziter Bestandteil der iterativen Evaluation der verschiedenen Designschritte.	Usability Testing (Test der Usability)	Impliziter Bestandteil des Prototypings (begrenzt auf die Konzeptebene)	Usability Evaluation (Prüfung der Usability)	Usability Inspection (Usability Untersuchung)
Iterative Screen Design Standards Evaluation (Iterative Verbesserung der Screen Design Standards)	Impliziter Bestandteil des Prototypings.	Nicht Bestandteil von Contextual Design.	Impliziter Bestandteil des Prototypings.	Impliziter Bestandteil des Prototypings.
Style Guide Development (Entwicklung eines Produkt Style Guides) Die Erstellung eines Styleguides im Sinne von Mayhew (Produkt Style Guide) einzigartig!	Nicht vorhanden!	Nicht vorhanden!	Nicht vorhanden!	Standards and Style Definition (Erstellung eines Styleguides) Weniger umfassend als Styleguide bei Mayhew!
Detailed User Interface Design (Detailliertes Design der Benutzerschnittstelle)	Impliziter Bestandteil des Prototypings.	Nicht Bestandteil von Contextual Design.	Impliziter Bestandteil der vorhandenen Designschritte.	Impliziter Bestandteil des Prototypings.
Implizit in übrigen Designaktivitäten vorhanden.	Implizit in übrigen Designaktivitäten vorhanden.	Implizit in übrigen Designaktivitäten vorhanden.	Implizit in übrigen Designaktivitäten vorhanden.	Operational Contextualization (Kontextanpassung)
Iterative Evaluation des detaillierten User Interface Designs Dieser Schritt mündet im fertigen System (evolutionärer Prototyp).	User Interface Implementation (Implementation der Benutzerschnittstelle)	Nicht Bestandteil von Contextual Design.	Nicht Bestandteil von Scenario Based Design.	Prioritized Concentric Construction (Konzentrische Entwicklung)
Bestandteil von Iterative Detailed User Interface Design Evaluation.	Nicht Bestandteil von Delta.	Nicht Bestandteil von Contextual Design.	Nicht Bestandteil von Scenario Based Design.	Architectural Iteration (Iterativer Architekturentwurf)

Tabelle11: Vergleich von 5 bekannten Usability Engineering Prozessmodellen – Fortsetzung.

Usability Engineering Lifecycle (Mayhew 1999)	Delta (WM-data AB & Ericsson Radio Systems AB 2000)	Contextual Design (Beyer und Holtzblatt 1998)	Scenario Based Design (Rosson und Carroll 2002)	Usage-Centered Design (Constantine und Lockwood 1999)
Iteration bei der Entwicklung des detaillierten User Interfaces impliziert wiederholte Usability Tests.	Nach Implementation ist kein weiterer Usability Test vorgesehen.	Nicht Bestandteil von Contextual Design.	Impliziter Bestandteil des Prototypings.	Usability Inspection (Usability Untersuchung)
Nicht Bestandteil des Usability Engineering Lifecycles.	Nicht Bestandteil von Delta.	Nicht Bestandteil von Contextual Design.	User Documentation (Erstellung eines Benutzungsdokumentes)	Help System and Documentation (Erstellung einer Benutzerdokumentation und eines Hilfesystems) Alle Designschritte übergreifende Aktivität.
User Feedback (Einholen von Benutzerfeedback)	Nicht Bestandteil von Delta.	Nicht Bestandteil von Contextual Design.	Nicht Bestandteil von Scenario Based Design.	Nicht Bestandteil von Usage-Centered Design.

Tabelle11: Vergleich von 5 bekannten Usability Engineering Prozessmodellen – Fortsetzung.

