

D i p l o m a r b e i t

Community Driven Development

Einbeziehung von Nutzer-Communities in den Software-
Designprozess

Autor: Sinja Offenberg
MatrikelNr: 621858

Betreuer: Dipl.-Inf. Jan Heß
Erstprüfer: Prof. Dr. Volker Wulf
Zweitprüfer: Prof. Dr. Volkmar Pipek

Eingereicht: 21. Februar 2008

Abstract

Die Einbeziehung von Anwendern in den Designprozess von Software-Produkten ist immer stärker gefragt. Seit den 1970er Jahren steigt das internationale Interesse am Participatory Design. Auch Unternehmen haben erkannt, dass die Integration der Nutzer in die Entwicklung viele Vorteile mit sich bringt. Nutzer-Communities verbinden viele interessierte Nutzer, die sich oft gut mit den Produkten auskennen und bereitwillig ihr Wissen einbringen. Somit scheinen gerade diese Gemeinschaften eine gute Grundlage für die Nutzer-Partizipation zu bieten. Im Rahmen dieser Arbeit wurde ein neues Konzept entwickelt, um Nutzer-Communities strukturiert in Software-Designprozesse einzubeziehen. Das Community Driven Development sieht eine Integration der Anwender über den gesamten Entwicklungsprozess vor. Entwickler und Anwender erstellen gemeinsam das Pflichtenheft für die Projektentwicklung, wobei die Vertreter der Community im zentralen Entscheidungsgremium die Mehrheit bilden. In einer Fallstudie wurde dieses Konzept auf seine praktische Anwendbarkeit geprüft. Dazu wurden sowohl eine qualitative Analyse anhand von Beobachtungen und Interviews als auch eine quantitative Bewertung mittels einer produktintegrierten Feedbackmöglichkeit durchgeführt. Mehrere Maßnahmen zur Unterstützung und Optimierung des Ansatzes wurden entwickelt. Trotz einiger Schwierigkeiten, vor allem im Bereich der Kommunikation und Zeitplanung, hat sich das Konzept des Community Driven Development als vielversprechend erwiesen. Sowohl Anwender als auch Entwickler waren letztlich überzeugt von dem neuen Modell.

Inhaltsverzeichnis

Abstract	1
Inhaltsverzeichnis	2
1 Einleitung	5
1.1 Einführung	5
1.2 Thematik.....	6
1.3 Aufbau der Diplomarbeit.....	7
2 Online-Communities	8
2.1 Definitionen.....	8
2.2 Anforderungen.....	10
2.2.1 <i>Motivation zur aktiven Beteiligung</i>	11
2.2.2 <i>Gemeinschaftsgefühl und Loyalität</i>	13
2.2.3 <i>Usability und Sociability</i>	14
2.2.4 <i>Anforderungen an Nutzer-Communities</i>	16
2.3 Klassifikation.....	17
2.3.1 <i>Klassifikation nach Armstrong und Hagel</i>	17
2.3.2 <i>Klassifikation nach Seufert, Moisseeva und Steinbeck</i>	18
2.4 Nutzer-Communities.....	19
3 Software-Design	21
3.1 Qualitätsanforderungen an Software-Produkte.....	21
3.1.1 <i>Anforderungen an die Implementierung</i>	21
3.1.2 <i>Anforderungen an die Funktionalität</i>	22
3.1.3 <i>Benutzerspezifische Anforderungen</i>	22
3.2 Modelle der Software-Entwicklung	24
3.2.1 <i>Wasserfallmodell</i>	24
3.2.2 <i>Iteratives Phasenmodell</i>	25
3.2.3 <i>Rapid Prototyping</i>	25
3.2.4 <i>Spiralmodell</i>	25
3.2.5 <i>V-Modell</i>	26
3.2.6 <i>STEPS-Modell</i>	27
3.2.7 <i>Design-Use-Cycle</i>	27
3.3 Design-Konzepte mit Einbeziehung der Nutzer	28
3.3.1 <i>User Centered Design</i>	28

3.3.2	<i>Participatory Design</i>	28
3.3.3	<i>Open Innovation</i>	29
3.3.4	<i>User Driven Development</i>	29
3.3.5	<i>Community Driven Development</i>	30
3.4	Vorteile und Probleme der Nutzerpartizipation	31
3.5	Techniken zur Nutzereinbindung	33
3.5.1	<i>Beobachtungen und Interviews</i>	34
3.5.2	<i>Usability-Untersuchungen</i>	35
3.5.3	<i>Probes</i>	35
3.5.4	<i>Workshops, Gruppendiskussionen</i>	35
3.5.5	<i>Innovation Play</i>	36
3.5.6	<i>Rapid Prototyping</i>	36
3.5.7	<i>Participatory Programming</i>	36
4	Einbeziehung einer Nutzer-Community in den Designprozess	37
4.1	Fallbeispiel Kite Surfing – Lead User Theorie.....	37
4.2	Studien nach Reichwald und Piller – interaktive Wertschöpfung.....	39
4.3	Fallbeispiel Audi – Community Based Innovation.....	41
4.4	Fallbeispiel Idavall – UDD in der Software-Entwicklung	45
5	Community Driven Development	46
5.1	Motivation	46
5.2	CDD – Konzept der strukturierten Community-Beteiligung.....	47
5.3	Forschungsfragen.....	48
6	Fallstudie Omega	50
6.1	Überblick über das Anwendungsfeld	50
6.1.1	<i>Omega-Community</i>	50
6.1.2	<i>OmegaTV</i>	51
6.1.3	<i>Vorgehensweise zur Entwicklung von OmegaTV4</i>	52
6.2	Untersuchungsdesign	56
6.3	Methodik und Interviewleitfäden	57
6.4	Beobachtungen der ersten Projektphase.....	58
6.5	CDD-Unterstützung	62
6.6	Beobachtungen der zweiten Projektphase	68
6.7	Beurteilung des CDD-Ansatzes	74
6.7.1	<i>Quantitative Bewertung des Gesamtergebnisses</i>	74
6.7.2	<i>Qualitative Beurteilung des Ansatzes</i>	78

6.7.3 Probleme und Verbesserungspotentiale	79
7 Diskussion.....	85
7.1 Motivation	85
7.2 CDD-Unterstützung	88
7.3 CDD-Konzept	91
8 Fazit.....	95
9 Zusammenfassung und Ausblick.....	97
9.1 Zusammenfassung.....	97
9.2 Ausblick	99
Abkürzungsverzeichnis	100
Abbildungsverzeichnis	101
Literaturverzeichnis	102
Anhang.....	108
Erklärung.....	109

1 Einleitung

“Developing software systems without listening to the users may be possible, but it is unlikely that such system will live up to the users’ requirements. Only the feedback of the users can ensure that users get what they want” [Hans02, S. 18].

1.1 Einführung

Eine Ausrichtung auf und Einbeziehung von Endnutzern findet im Software-Design schon seit Langem statt. Erste partizipative Ansätze gab es bereits in den 1970er Jahren in Skandinavien [vgl. AIEd+01, Levi98]. Angetrieben wurde diese Entwicklung durch Gewerkschaften, die für mehr Demokratie am Arbeitsplatz eintraten und den Arbeitern somit mehr Einflussnahme auf ihre Arbeitssysteme verschaffen wollten. Ihr Ziel dabei war es, sicherzustellen, dass die Arbeiter im Zuge der Computerisierung nicht ersetzt, sondern ihre Fähigkeiten genutzt und durch passende Systeme ideal unterstützt wurden. Eine grundlegende Annahme dabei war, dass Technologien manchen Personen mehr Vorteile bieten als anderen und dass diejenigen, die nicht an deren Designprozess beteiligt sind, in dieser Hinsicht das Nachsehen haben. Somit forderten die Gewerkschaften, die Arbeiter in die Systementwicklung einzubeziehen, um ihre Arbeitsweise und Kenntnisse besser unterstützen zu können. Auch in den 1980er Jahren waren es überwiegend skandinavische Forscher, die die Entwicklung auf diesem Gebiet vorantrieben. Doch auch in Europa und den USA stieg das Interesse am Participatory Design (PD), in immer mehr internationalen Konferenzen arbeiten Forscher aus verschiedenen Ländern zusammen. Zu den erfolgreichsten Konferenzen zählen die *Participatory Design Conference*, die seit 1990 alle zwei Jahre an wechselnden Orten in den USA und Europa stattfindet, sowie die seit dem Jahr 2000 ebenfalls zweijährig in den skandinavischen Ländern veranstaltete *NordiCHI*¹.

Mit den Jahren nahm die Zahl an unterschiedlichen PD-Techniken immer mehr zu, wobei der Einfluss der Anwender ebenfalls anstieg. Anfangs wurden die Nutzer eher

¹ NordiCHI = Nordic Conference on Human-Computer Interaction

passiv durch Beobachtungen und Interviews studiert, mit der Zeit entwickelten Forscher jedoch viele weitere Möglichkeiten, die Anwender aktiv einzubeziehen, sie sogar als Co-Designer zu betrachten. Auch viele Unternehmen haben mittlerweile erkannt, dass nur mit Hilfe der zukünftigen Anwender eine langfristig erfolgreiche Software entwickelt werden kann. Mit der Einbeziehung der Nutzer in Design und Entwicklung eröffnen sich viele Möglichkeiten. Zum Einen erhalten die Unternehmen kostengünstig Unterstützung und neue Ideen aus Sichtweisen, die ihnen selbst oft nur vage bekannt sind. Zum Anderen bindet das den Anwendern entgegen gebrachte Vertrauen diese verstärkt an Unternehmen und Produkt.

Online-Communities verbinden Personen mit gleichen Interessen zum gemeinsamen Gedankenaustausch. Gerade auf diesem Weg bietet das Internet somit auch Firmen die Möglichkeit, schnell und effizient mit ihren Kunden in Kontakt zu treten. Heutzutage haben viele Unternehmen auf ihren eigenen Internetseiten eine Plattform geschaffen, über die sich die Anwender sowohl untereinander als auch mit Repräsentanten der Firma austauschen können. Oft befassen sich einige Mitglieder solcher Nutzer-Communities sehr intensiv mit den Produkten und sind somit potentielle Lieferanten für innovative Ideen zur Weiterentwicklung. Aus diesem Grund scheinen gerade virtuelle Nutzer-Communities gut dafür geeignet, eine große Zahl zukünftiger Anwender intensiv in Software-Designprozesse einzubeziehen.

1.2 Thematik

Zwar existieren mehrere Konzepte zur Einbeziehung der Anwender ins Software-Design, keines von diesen nutzt jedoch die Vorteile, die eine etablierte Nutzer-Community zu diesen Zwecken bieten kann. Deshalb wurde im Rahmen dieser Arbeit ein neues Konzept entwickelt, das es ermöglicht, nicht nur einzelne Anwender sondern ganze Nutzer-Communities strukturiert in den Software-Entwicklungsprozess einzubeziehen. Das entwickelte Konzept wurde zusätzlich in einem konkreten Anwendungsfall evaluiert. Begleitet wurde dazu ein Software-Unternehmen bei der Entwicklung einer neuen Produkt-Version unter Einbeziehung der zukünftigen Nutzer des Systems. Zu diesem Zweck stellte das Entwickler-Team gemeinsam mit den Anwendern ein Pflichtenheft zusammen. Neben der beobachtenden Begleitung des Projektverlaufs wurde anhand von Interviews, sowohl

mit Anwendern als auch mit Entwicklern, eine Anforderungsanalyse für die Herangehensweise an ein solches Projekt durchgeführt. Im Mittelpunkt der Untersuchungen stand die Erfassung der Motivation aller Beteiligten, ihrer Erwartungen und Vorbehalte an das Projekt, der praktischen Möglichkeiten zur Einbeziehung von Anwendern sowie der in der Praxis auftretenden Schwierigkeiten. Von zentraler Bedeutung war die Herausarbeitung der Vorteile, die die Einbeziehung einer Nutzer-Community gegenüber Einzelanwender bietet. Neben der qualitativen Analyse wurde zusätzlich eine quantitative Auswertung des Konzepts über einen in das Produkt integrierten Bewertungsmechanismus vorgenommen. Aus den gesammelten Erkenntnissen wurde ein Maßnahmenkatalog zur Unterstützung der Nutzerartikulation abgeleitet. Nachdem eingangs eine Beschränkung auf den konkreten Fall stattfand, wurden abschließend Verbesserungsvorschläge für allgemeine Anwendungsfälle abgeleitet.

1.3 Aufbau der Diplomarbeit

Das Forschungsfeld der communityorientierten Nutzerbeteiligung wird im Rahmen dieser Arbeit aus zwei unterschiedlichen Richtungen angegangen. In einer theoretischen Untersuchung wird zunächst der aktuelle Stand der Wissenschaft erarbeitet. Dieser Teil der Diplomarbeit umfasst die Themen Online-Communities (Kapitel 2) und Software-Design (Kapitel 3), darunter speziell die Methoden zur Einbeziehung von Nutzern in Designprozesse (Kapitel 4). Zum Abschluss der theoretischen Betrachtung wird ein neues Konzept zur Einbeziehung von Nutzer-Communities vorgestellt (Kapitel 5). In diesem Zusammenhang ergibt sich eine Reihe interessanter Forschungsfragen, die anhand eines praktischen Anwendungsfalls untersucht werden (Kapitel 6). Bei der Evaluierung des Konzepts werden sowohl Potentiale des Ansatzes als auch Schwierigkeiten bei dessen Umsetzung herausgearbeitet. Des Weiteren wird im Rahmen der Projektbegleitung eine produktintegrierte Feedbackunterstützungsmaßnahme konzipiert und implementiert, die Rückschlüsse auf den bisherigen Erfolg oder Misserfolg der Umsetzung liefert. Abschließend erfolgt eine Bewertung des Konzepts basierend auf den im Vorfeld identifizierten Forschungsfragen und den gesammelten Ergebnissen (Kapitel 7). Die Diskussion beinhaltet die Identifikation von Verbesserungspotentialen und Richtlinien für zukünftige Anwendungen des Konzepts.

2 Online-Communities

Im Zuge der weltweiten Vernetzung wird immer häufiger der Begriff der *Online-Community* beziehungsweise das Synonym der *virtuellen Community* verwendet. Gemeint war damit anfangs die Gemeinschaft aller durch das Internet verbundenen Menschen, die sich im Gegensatz zu sonstigen Gruppierungen nicht zwangsläufig körperlich, sondern online im virtuellen Raum begegnen. Diese Definition ist jedoch veraltet und wie Dr. Helmut Merz feststellt, wird der Begriff mit zunehmender Nutzung immer undeutlicher: *„Gerade wenn es darum geht, für ein Produkt mit dem Einsatzbereich Internet zu werben, scheint Community ein verkaufsförderndes Merkmal darzustellen, ohne dass dessen Bedeutung überhaupt klar wäre“* [Merz02].

Aufgrund dieser Unklarheit werden im Folgenden zunächst die bekanntesten Definitionen für virtuelle Communities verglichen und die Anforderungen an solche Gemeinschaften dargelegt. Anschließend werden unterschiedliche Möglichkeiten zur Klassifikation von Online-Communities vorgestellt. Im vierten Unterkapitel wird die Form der Nutzer-Communities genauer untersucht.

2.1 Definitionen

Der englische Begriff *Community* bedeutet übersetzt *„Gemeinschaft, Gemeinde, Körperschaft, Gemeinwesen; dies kann sowohl geografisch interpretiert werden als eine Gruppe von Menschen, die in derselben Region oder am selben Ort leben, oder sich auf Menschen beziehen, die zusammenleben und gemeinsame Interessen haben oder an einer gemeinsamen Aufgabe arbeiten“* [Merz02].

Die angloamerikanische Gesellschaft verwendet diesen Ausdruck schon seit Langem in der Soziologie und verbindet damit soziale Unterstützungsnetzwerke wie Sozialhilfe, Sozialarbeit oder auch Nachbarschaftshilfe und Spendenaktionen [vgl. SeMo+02, S. 2]. Im Deutschen wird einerseits durch die Zunahme von Anglizismen (vor allem im IT-Bereich), andererseits auch durch die historisch bedingte Ablehnung des Gemeinschaftsbegriffes ebenfalls auf den englischen Ausdruck zurückgegriffen [vgl. SeMo+02, S. 2-3].

Bei einer *Online-Community* begegnen sich die Mitglieder im virtuellen Netz, die räumliche Beschränkung ist somit aufgehoben, wodurch das gemeinsame Interesse in den Vordergrund der Gemeinschaftsverbundenheit tritt. Der Autor Derek M. Powazek erläuterte in einem Interview im August 2001: *“It’s not enough to be a member of a virtual community if you’ve got nothing in common with anyone there”* [Finc01]. In seinem Buch erklärt er jedoch, es gebe keine allgemein anerkannte Definition für virtuelle Communities, jeder habe seine eigene Interpretation, was die Verbundenheit zu einer solchen Gemeinschaft bedeute [vgl. Powa01, Vorwort, S. xx]. Lazar und Preece bestätigen dies: *„We all have our own notion of what an online community is. It isn’t hard to understand, but it is slippery to define and tricky to measure“* [LaPr02, S. 3].

Mit der Zeit und unter Einflüssen aus verschiedensten Forschungsgebieten haben sich somit immer wieder neue Definitionen des Begriffs entwickelt. Der Sozialwissenschaftler Howard Rheingold gilt als eine der ersten und führenden Autoritäten im Bereich virtueller Communities. 1993 prägte er den Ausdruck in seinem Buch *The Virtual Community*: *„Virtual communities are social aggregations that emerge from the Net when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace“* [Rhei93, Introduction]. Powazek stellt in [Powa01] ebenfalls seine eigene Definition vor, nach der Online-Communities entstehen, wenn Nutzern Werkzeuge gegeben werden, mittels derer sie ihre Meinung öffentlich und unmittelbar zum Ausdruck bringen können, wodurch sich über längere Zeit hinweg vertraute Beziehungen entwickeln [vgl. Powa01, Vorwort, S. xxii].

Mittelpunkt der Diskussion beziehungsweise des gemeinsamen Interesses kann dabei vieles sein. So schreibt Powazek weiterhin: *“The web is now a part of everything, and communities form around people with like interests. Sometimes, those communities form around local organizations (like a church). Sometimes, those communities form around an interest (say, golf) or a demographic (parents, teenagers, elderly). And, sometimes, they form around a product”* [Powa01, S. 219/220].

Für den Community-Aspekt von Bedeutung sind somit der Austausch im virtuellen Netz, ein gemeinsames Interesse sowie ein Gemeinschaftsgefühl [vgl. KaCh+00, S. 416].

2.2 Anforderungen

Die Anzahl und Vielfalt virtueller Communities steigt immer mehr an [vgl. LaPr02, S. 9]. Über das Internet kann jeder schnell und ohne große Umstände verschiedenen Gemeinschaften beitreten. Zwar gibt es bei allen Communities Einschränkungen zur Mitgliedschaft, jedoch existieren so viele virtuelle Gemeinschaften, dass Personen meist Mitglied in mehreren Communities mit oft sehr unterschiedlichen Zielen sind.

Doch wie entstehen virtuelle Communities und was ist erforderlich, um sie langfristig zu erhalten? Powazek erläutert in [Powa01] ausführlich die einzelnen Schritte, die zum Aufbau einer Online-Community nötig sind. Basis jeder virtuellen Community ist ein Kommunikationsmedium wie beispielsweise ein E-Mail-Verteiler oder in vielen Fällen ein Forum. Weitere Grundvoraussetzungen sind eine aktive Nutzergemeinschaft sowie ein gemeinsames Interesse aller Mitglieder. Das Gemeinschaftsgefühl, welches für den langfristigen Bestand einer virtuellen Community von großer Bedeutung ist, bildet sich erst mit fortlaufender Nutzung der Kommunikationsplattform. Lazar und Preece befassen sich in [LaPr02] damit zu untersuchen, worin einzelne Personen den Erfolg einer Community sehen. So sei aus Gründersicht eine konstante Nutzerschaft und Nutzung der Community der entscheidende Faktor, für den Community-Leiter hingegen die Anerkennung seiner Rolle sowie eine hohe Anzahl an eigenen Beiträgen. Für Moderatoren seien glückliche Community-Mitglieder, die sich angemessen verhalten, der wichtigste Aspekt, gemessen werden könne dies somit anhand der Anzahl der Beschwerden oder unpassender Postings. Die Mitglieder selbst sind mit ihrer Community zufrieden, wenn sie von ihr genau das erhalten, was sie erwartet haben, seien es Informationen, Freundschaften, Unterstützung oder anderes. Aus Sicht eines Managers sind eine höhere Anzahl an Seitenbesuchen, höhere Werbeeinnahmen, mehr Verkäufe oder auch die Bekanntheit und das Image der Marke entscheidend. Jedoch gibt es keinen allgemeingültigen Indikator, der den Erfolg einer Community messen lässt. „*A community is not necessarily successful because there are 500 posts a week, or*

because 3000 people are subscribed to the community, or because 100% of community members have posted a message. (...) There is not a single quantitative measurement that can determine success" [LaPr02, S. 29].

Allgemein als wichtige Faktoren für den dauerhaften Erhalt einer Community anerkannt sind jedoch deren *Usability* (Nutzbarkeit), *Sociability* (Gemeinschaftstauglichkeit), ein Gemeinschaftsgefühl sowie die Bereitschaft der Mitglieder zur aktiven Beteiligung [vgl. Pree00, LaPr02]. Diese Voraussetzungen werden im Folgenden genauer erläutert.

2.2.1 Motivation zur aktiven Beteiligung

Jennifer Preece nennt in [Pree00] unterschiedliche Gründe, die Nutzer dazu motivieren, Communities beizutreten. Dazu gehören unter anderem die Bedürfnisse, neue Freundschaften zu schließen, Interessen zu diskutieren, über Politik zu debattieren, Informationen zu erhalten, Wissen zu teilen, Empathie von Gleichgesinnten zu empfangen und bei der Lösung eines Problems unterstützt zu werden [vgl. Pree00, Chapter 2].

Die Motivation der Mitglieder, sich aktiv in ihrer Community zu beteiligen, ist von großer Bedeutung für den langfristigen Bestand der Gemeinschaft. Als *Lurker* werden solche Personen bezeichnet, die in einer Community hauptsächlich lesen, jedoch kaum selbst Beiträge schreiben. Diese stellen jedoch in den meisten Online-Communities den Großteil der Mitglieder dar [vgl. YuJi+07, S. 144]. Lurker sind allerdings nicht grundlegend schlecht für die Gemeinschaft. So erklären Lazar und Preece, dass viele Lurker sich selbst als Teil der Community sehen, jedoch oft im Themengebiet noch zu unerfahren sind, um sich an Diskussionen zu beteiligen. Somit bestehe die Hoffnung, dass sich auch diese Personen nach einer Eingewöhnungsphase aktiv einbrächten [vgl. LaPr02, S. 30].

Yu et al. ermitteln in [YuJi+07] die unterschiedlichen Gründe, die Community-Mitglieder dazu veranlassen können, ihr Wissen in eine Gemeinschaft einzubringen. Hauptsächlich basieren diese Motivationen auf Eigennutz, doch es gibt auch weitere Indikatoren wie Selbstlosigkeit, Gemeinschaftsgeist und moralische Grundsätze. Die

acht von den Forschern festgestellten Gründe sind aktives Lernen, Selbstaufwertung, Wechselseitigkeit des Wissensaustauschs, Steigerung des eigenen Ansehens, Freude daran anderen zu helfen, Selbstschutz, moralisches Engagement und die Weiterentwicklung der virtuellen Community. Die Studie von Yu et al. erfolgt anhand einer Community of Practice (CoP), also einer Gemeinschaft, deren Austausch zur gegenseitigen Unterstützung bei Arbeitsprozessen dient. Die Ergebnisse der Untersuchung lassen sich jedoch im Groben auf alle Arten virtueller Communities übertragen. So beruht die Steigerung des Ansehens bei einer CoP auf fachgebietsspezifischen Kenntnissen, während in anderen Online-Communities an diesem Punkt das Bedürfnis Freunde zu finden in den Vordergrund tritt. Handelt es sich um eine zielgerichtete Community wie eine CoP, so sind sowohl die persönliche Wertschätzung des einzelnen Mitglieds für dieses Ziel als auch die Erfolgchancen, die der Einzelne dessen Erreichung zumisst, bedeutend für die persönliche Motivation. Jedoch sind diese Werte nicht durch den Community-Anbieter zu beeinflussen. In ihrer Studie ermitteln Yu et al. allerdings vier Mechanismen, mittels derer die Mitwirkungsbereitschaft der Mitglieder von Seiten des Anbieters unterstützt werden kann. Diese sind die im Folgenden vorgestellten Maßnahmen: Kodifizierung, Identitätsmanagement, Normen virtueller Communities sowie Kommunikationsmethoden [vgl. YuJi+07, S. 146].

Kodifizierung ist der dokumentbasierte Austausch von Wissen, beispielsweise über FAQs und Datenbanken. Wichtig sind eine gute Strukturierung sowie die Sicherstellung der Qualität der Inhalte in solchen Verzeichnissen. Ebenso wie durch eine zuverlässige Suchfunktion in Foren können dadurch einerseits schnelle Hilfestellungen bei bekannten Problemen gegeben sowie andererseits lästige Wiederholungen vermieden werden [vgl. YuJi+07, S. 146].

Computervermittelte Kommunikation ist ein Grundbaustein virtueller Communities und somit bedeutend für die Motivation der Mitglieder. Die wahrgenommene Medien-effizienz kann sich bei verschiedenen Nutzern stark unterscheiden, da sie von den eigenen Erfahrungen abhängig ist. Dies betrifft Erfahrungen sowohl mit dem vermittelnden Kanal oder Medium als auch mit bestimmten Diskussionsthemen und Kommunikationspartnern [vgl. YuJi+07, S.146]. Des Weiteren wirken laut Yu et al.

die Kommunikationseffektivität und aktive Lernerfolge positiv auf die Wahrnehmung der Medieneffizienz ein und somit auch auf die Motivation der Nutzer.

Gruppennormen geben den Mitgliedern vor, wie sie sich innerhalb der Community verhalten sollen. Laut Yu et al. erzeugt eine Situation, in der strenge Gruppennormen die soziale Interaktion bestimmen und die Mitglieder daran glauben, dass sich auch andere Teilnehmer an diese Regeln halten, sowohl eine starke gegenseitige Einigkeit als auch ein Entgegenkommen der Mitglieder untereinander. Somit verringern effiziente Gruppennormen das Bedürfnis sich selbst zu schützen und erhöhen den Spaßfaktor, den Meinungs Austausch sowie die Bereitschaft anderen zu helfen [vgl. YuJi+07, S. 147].

Mechanismen für das Identitätsmanagement betreffen sowohl die persönliche als auch die soziale Identität. Wird beispielweise die durchschnittliche Bewertung, die die Beiträge des Nutzers erhalten haben, in seinem Profil angezeigt, so spiegelt dies seine persönliche Identität wieder und verstärkt seine Motivation gute Beiträge zu liefern, um sein Ansehen zu verbessern. Die soziale Identität hingegen zeigt sich darin, zu welchen Nutzergruppen sich der Einzelne zuordnet. Einige Personen sind eher geneigt, anderen zu antworten und zu helfen, wenn diese zu den gleichen Interessengruppen gehören [vgl. YuJi+07, S. 147].

Neben dem Ansehen durch andere ist jedoch auch das Selbstbild von Bedeutung. So verstärkt die Erkenntnis, dass die eigene Einbringung die Community ihrem Ziel näher bringt, sowohl Selbstachtung als auch Selbstvertrauen. Selbstbewusste Personen sind oft eher dazu bereit, anderen zu helfen. Des Weiteren bringt das Nachdenken über Probleme anderer, denjenigen selbst eventuell auch voran und ermöglicht ihm neue Sichtweisen [vgl. YuJi+07, S. 148].

2.2.2 Gemeinschaftsgefühl und Loyalität

“Ein wichtiger Aspekt ist auch die Festigkeit der Bindung der Mitglieder an ihre Community” [Merz02]. Wie hier Dr. Helmut Merz betonen viele Experten immer wieder die Bedeutung der starken Beziehungen, die sich über längere Zeit in einer Community aufbauen [vgl. Rhei93, KaCh+00, Pree00, Powa01].

Intimität ist laut Powazek eine notwendige Komponente, um virtuelle Communities zu bilden. Aus diesem Grund sei es die Aufgabe der Web-Designer, einen Ort zu erschaffen, an dem sich die Nutzer wohl fühlen, über den Computer als emotionslose Maschine hinweg sehen können und auch in der virtuellen Umgebung beginnen zu fühlen [Powa01, S. 142]. Ein so entstehendes Gemeinschaftsgefühl verstärkt die Verbundenheit zur Community sowie deren Mitgliedern und erhöht die Bereitschaft sich aktiv zu beteiligen. Besonders diese emotionale Verbindung ist ein Anreiz für viele Unternehmen, Communities für ihre Zwecke einzubinden [vgl. Hans02, S. 1]. Unternehmen sehen darin die Möglichkeit, die Loyalität der Mitglieder zur Community auf die Kunden-Anbieter-Beziehung zu übertragen. So ermittelte AT&T² bei der Überwachung von Aktivitäten innerhalb einer virtuellen Gemeinschaft, „dass Community-Mitglieder wesentlich länger Kunden bleiben“ [Merz02].

Somit ist die emotionale Bindung an die Gemeinschaft sowohl bei kommerziellen als auch bei Communities ohne finanzielle Interessen sehr wichtig für deren langfristigen Fortbestand.

2.2.3 Usability und Sociability

Durch ihre Mitglieder organisieren und verwalten sich virtuelle Communities selbst. Es ist jedoch notwendig, dass ein Rahmenwerk geschaffen wird, welches die evolutionäre Entwicklung einer Community unterstützt [vgl. Wied06, S. 22]. *“Success of an online community is encouraged by a blend of well-designed software (i.e., usability) and carefully crafted social policies (i.e., sociability)”* [LaPr02, S. 15]. Usability bezieht sich somit auf die technischen Grundlagen und kann durch den Anbieter kontrolliert werden. Im Gegensatz dazu ist Sociability darauf bezogen, wie die Mitglieder der Community miteinander umgehen. Sie kann also nicht vom Anbieter kontrolliert, jedoch stark durch ihn beeinflusst werden [vgl. LaPr02, S. 2, 41].

Usability

Nach Preece sind die Grundfunktionen kommunikationsvermittelnder Softwaresysteme der Informationsaustausch, die Bereitstellung von Unterstützung, die

²American Telephone & Telegraph Corporation – nordamerikanischer Telekommunikationskonzern

Schaffung informeller Interaktion und die Unterstützung bei der Diskussion von Ideen [vgl. Pree00]. Diese Funktionalitäten sollten möglichst benutzerfreundlich gestaltet sein, um schnelles Lernen, geringe Fehlerquoten und eine hohe Produktivität zu ermöglichen [vgl. LaPr02, S. 8].

Shneiderman formulierte 1998 drei generelle Prinzipien für Software-Usability: Konsistenz, Kontrolle und Vorhersagbarkeit [vgl. Shne98]. Konsistenz ist sowohl bei Design und Begriffen als auch bei Abläufen erforderlich. Nutzer wünschen sich ein System, das ihnen die Kontrolle über seine Funktionen überlässt statt sie einzuschränken. Ist eine Software sowohl konsistent als auch kontrollierbar, so ist sie ebenfalls vorhersehbar, das heißt, Nutzer können auf ihren Erfahrungen mit dem System aufbauen. Durch diese drei Prinzipien wird der Software-Gebrauch sowohl einfach als auch zufriedenstellend für den Anwender [vgl. LaPr02, S. 2].

Sociability

Im Idealfall würden sich alle Mitglieder einer Community auch bei Diskussionen freundlich und höflich verhalten sowie die Bedürfnisse der anderen Nutzer beachten [vgl. LaPr02, S. 7]. Die Anonymität des Internets verleitet allerdings immer wieder einige Querulanten dazu, Unfrieden innerhalb der Gemeinschaft zu stiften. Um dem entgegenzuwirken, ist es Aufgabe der Anbieter, Richtlinien festzulegen und für deren Einhaltung zu sorgen. *„Sociability focuses on social interaction. Communities with good sociability have social policies that support the community’s purpose and are understandable, socially acceptable, and practicable”* [LaPr02, S. 15 in Anlehnung an Pree00]. Viele dieser Regeln sind weit verbreitet und den Nutzern allgemein bekannt, andere werden speziell für bestimmte Communities aufgestellt [vgl. LaPr02, S. 7].

Eine weitere wichtige Voraussetzung für eine virtuelle Community ist die deutliche Darstellung und klare Formulierung ihres Themas beziehungsweise Ziels. Diese dient vor allem dazu, neue Mitglieder anzuziehen und der Gemeinschaft eine Grundlage für ihre Verbundenheit zu geben [vgl. LaPr02, S. 17]. Weitere bedeutende Aspekte sind eine gute Qualität der Inhalte sowie die Absicherung privater Angaben vor Fremdzugriffen [vgl. KaCh+00, S. 419-420], denn Vertrauen und Sicherheit sind in jeder Online-Community notwendig, um den Anwendern das Gefühl zu vermitteln,

dass ihre Privatsphäre geschützt ist, wodurch sie freier miteinander kommunizieren können [vgl. LaPr02, Pree00]. Zu diesem Zweck ist es sinnvoll, auf der Homepage eine Datenschutzvereinbarung zu veröffentlichen. Ebenfalls abgesichert sein sollten die Erreichbarkeit der Community-Plattform sowie Backups der Beiträge [vgl. LaPr02, S. 36-37].

2.2.4 Anforderungen an Nutzer-Communities

Im Gegensatz zu anderen virtuellen Gemeinschaften ist bei einer Nutzer-Community die Zugehörigkeit und Dauerhaftigkeit dieser Verbindung ausschließlich von dem Interesse und Willen des Kunden bestimmt [vgl. MiMä05, S. 34]. In Konkurrenz mit anderen Communities ist es Aufgabe des Anbieters, die Attraktivität seiner Seite für den Kunden möglichst hoch zu halten.

Mittilä und Mäntymäki suchen in [MiMä05] nach Ursachen, die dazu führen, dass Nutzer-Communities an Attraktivität für die Anwender verlieren. Dabei unterteilen sie diese Gründe in die vier Kategorien *Inhalte*, *Mitgliedschaften*, *Kultur* sowie *Usability* [vgl. MiMä05, S. 32]. Zu den inhaltlichen Faktoren zählen Themen und Diskussionsqualität. Themen können beispielsweise uninteressant werden, wenn das Ziel, auf das sie sich beziehen, erreicht wurde. Die Qualität der Diskussion nimmt ab, wenn sich Fragen ständig wiederholen, provozierende Inhalte verbreitet werden oder ein Mangel an neuen Ideen und Überlegungen herrscht. Eine gute Moderation kann die Diskussionsqualität erhöhen. Das Moderatorenverhalten gehört zusammen mit den Regeln und der Atmosphäre innerhalb einer Community zum Bereich der Kultur. Die Kategorie Mitgliedschaften bezieht sich auf die Community-Mitglieder und deren Beziehungen zueinander. So können Streitigkeiten unter einzelnen Community-Mitgliedern ebenfalls zur Abwendung von einer Community führen wie das Fehlen eines Gemeinschaftsgefühls. Usability steht für sowohl technische als auch nutzerspezifische Kompetenzen. Zu den technischen Problemen gehören zum Beispiel eine zu langsame Verarbeitung und hohe Komplexität der Community-Plattform.

2.3 Klassifikation

Es existieren viele Online-Communities mit äußerst unterschiedlichen Zwecken [vgl. LaPr02, S. 9]. Um einen Überblick über die Vielfalt der möglichen Varianten zu geben, werden im Folgenden verschiedene Klassifikationsmethoden vorgestellt. Online-Communities könnten beispielsweise danach klassifiziert werden, in welcher Beziehung sie zu einer physikalischen Gemeinschaft stehen [vgl. LaPr02, S. 5]. Diese Art der Klassifikation orientiert sich somit an der Häufigkeit von Face-to-Face-Begegnungen, eine andere an der Art der verwendeten Kommunikationssoftware [vgl. LaPr02, S. 5-6]. Die gebräuchlichsten Kategorisierungen richten sich jedoch nach dem Zweck der Community [vgl. Chan04, S. 12], zwei von diesen werden hier präsentiert.

2.3.1 Klassifikation nach Armstrong und Hagel

Eine der bekanntesten Klassifikationen virtueller Communities ist die von Arthur G. Armstrong und John Hagel, die 1995 vier verschiedene Typen von Online-Gemeinschaften erkannten [vgl. KaCh+00, ArHa95]. Es handelt sich dabei um transaktions-, interessen-, phantasie- und beziehungsorientierte Communities. *Communities of Transaction* dienen zur wirtschaftlichen Leistungserstellung durch gemeinsame Transaktionen. Dies ist beispielsweise bei Online-Versandhäusern wie Amazon³ der Fall. Bei *Communities of Interest* findet ein Informationsaustausch rund um ein bestimmtes gemeinsames Interesse statt. Oft werden dafür Newsgroups und Diskussionsforen genutzt. *Communities of Fantasy*, zum Beispiel virtuelle Welten, dienen der Unterhaltung durch Phantasie und Spiel. Bei *Communities of Relationship* steht die soziale und emotionale Beziehung der Mitglieder im Vordergrund. Somit dienen sie dem gegenseitigen Austausch und der Unterstützung. Beispielsweise existieren viele solche Communities für Personen, die an einer gemeinsamen Krankheit leiden, und ähneln dadurch Selbsthilfegruppen.

³ <http://www.amazon.de>

2.3.2 Klassifikation nach Seufert, Moisseeva und Steinbeck

Aufbauend auf der zuvor dargestellten Variante beruht die Klassifikation nach Seufert, Moisseeva und Steinbeck ebenfalls auf dem „zielsetzenden Bereich, den eine Gemeinschaft fokussiert“ [SeMo+02, S. 7]. Erweitert um den Lern-Aspekt werden die Communities in die Kategorien *Arbeit*, *Forschung*, *Lernen* und *privates Interesse* unterteilt.

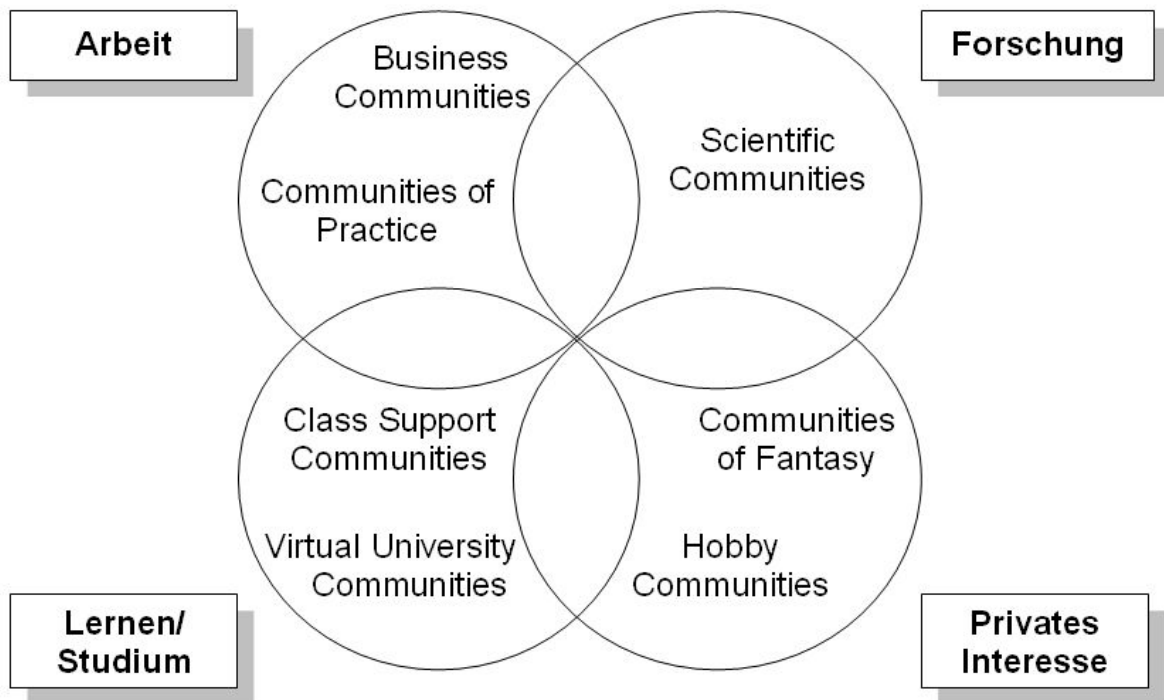


Abbildung 1: Klassifikation von Communities nach Seufert et al. [vgl. SeMo+02]

Hobby Communities entsprechen den von Hagel und Armstrong definierten *Communities of Interest*. Stark auf Wissensaustausch und gemeinsames Lernen aufbauend sind *Virtual University Communities* und *Class Support Communities*. Die bekanntesten Gemeinschaften sind *Communities of Practice*, die auch als *Problem Solving Virtual Communities* bezeichnet werden. Nach Seufert et al. stehen im Zentrum ihrer Aktivitäten der Austausch von Ideen, Einsichten und Erkenntnissen, das gemeinsame Lernen sowie die gegenseitige Unterstützung am Arbeitsplatz [vgl. SeMo+02, S. 8]. Jedoch ist es nicht Ziel der Mitglieder, gemeinsam Aufgaben zu lösen, sondern gemeinsames Wissen zu teilen, um getrennt voneinander Probleme zu lösen [vgl. Wied06, S. 20]. *Business Communities* hingegen verfolgen finanzielle Interessen und dadurch gemeinsame wirtschaftliche Ziele [vgl. SeMo+02, S. 8]. Im

Bereich der Forschung existieren *Scientific Communities*, die sich mit dem Austausch und Diskurs wissenschaftlicher Ergebnisse befassen [vgl. SeMo+02, S. 8].

2.4 Nutzer-Communities

1997 waren es erneut Hagel und Armstrong, die im Zuge der zunehmenden Entwicklung im Bereich des E-Commerce eine neue Perspektive entdeckten. *„We believe that commercial success in the on-line area will belong to those who organize virtual communities to meet multiple social and commercial needs. By creating strong virtual communities, businesses will be able to build membership audiences and use those audiences to bring in revenues in the form of advertising, transaction fees, and membership fees“* [HaAr97].

Barry Wellman beschreibt in [Well98] *Communities of Consumers* als Online-Diskussionsrunden aus Kunden, Mandanten und Anbietern einer Organisation, die das Unternehmen mit Informationen über die Eigenschaften seiner Kunden und Mandanten sowie Wissen über deren Wunschvorstellungen versorgen und gleichzeitig Loyalität zu den Produkten der Organisation fördern können. Doch um Erfolg aus diesen Informationen zu generieren und langfristig die Kundenbindung zu garantieren, muss das Unternehmen entsprechend handeln. Zum Einen müssen die Bedürfnisse der Nutzer erkannt und mittels Usability und Sociability ein gutes Rahmenwerk für die Community-Plattform geschaffen werden. Zum Anderen sollte die Organisation die gesammelten Informationen für die weitere Produktentwicklung nutzen. Denn oft bringen Nutzer, die sich verstärkt mit den Produkten und der zugehörigen Community befassen, viele innovative Ideen für Verbesserungen oder neue Produkte in die Diskussionen ein [vgl. FüBa+06, S. 58]. Dr. Helmut Merz sichert gerade denjenigen Unternehmen großen Erfolg zu, die ihre Nutzer kennen und *„denen es gelingt, eine klar wahrnehmbare Positionierung, d.h. eine klare Ausrichtung auf bestimmte Zielgruppen und deren Bedürfnisse zu erreichen“* [Merz02]. Sind diese Voraussetzungen geschaffen, so ist es möglich, die Zielgruppen präzise anzusprechen [vgl. Orma05, S. 167].

Ein weiterer bedeutender Vorteil von Nutzer-Communities ist, dass Produktanwender und Kaufinteressenten sich untereinander austauschen können.

Interessenten vertrauen viel eher auf die kritische Meinung anderer Nutzer als auf Anpreisungen durch Anbieter. Unternehmer wissen dies zu schätzen und sehen positive Kommentare der Anwender als kostengünstige und effektive Form des Marketings [vgl. LaPr02, S. 5]. So bieten Nutzer-Communities sowohl für Unternehmen als auch Anwender einige Vorteile. Denn auch die Anwender profitieren durch ihr gemeinsames Auftreten am Markt, welches ihnen mehr Einfluss auf Unternehmensentscheidungen verleiht [vgl. Orma05, S. 167].

Durch all diese Gründe scheint es bedeutend für Unternehmen, Anwender auch aktiv in ihre Produktplanungen mit einzubeziehen. Auch Füller et al. vermuten, dass Mitglieder einer Nutzer-Community besonders geeignet sind, um virtuell an der Entwicklung neuer Produkte teilzunehmen. Sie begründen dies sowohl mit dem hohen Interesse an und Wissen über Produkte als auch der Präsenz der Anwender im Internet [vgl. FÜBa+06, S. 58].

3 Software-Design

Nachdem die Software-Entwicklung zuvor als ein einzelner untrennbarer Vorgang galt, entwickelte Winston W. Royce 1970 das erste Vorgehensmodell, das den Prozess in mehrere Phasen unterteilt [vgl. Royc70]. Heutzutage gilt jedoch auch dieses Modell als veraltet, so betitelten Rauterberg et al. schon 1993 eines ihrer Paper „*Phasenmodell ist OUT*“ [RaMo+93]. Weiterhin erklärten sie die Benutzerbeteiligung als entscheidende Zukunftsperspektive, was viele andere Autoren ebenfalls bestätigen.

Im Folgenden werden zunächst die Anforderungen an gute Software-Produkte und einige Modelle der Software-Entwicklung vorgestellt. Anschließend folgt eine grobe Beschreibung unterschiedlicher Ansätze zur Nutzereinbeziehung, gefolgt von der genaueren Vorstellung einzelner Techniken.

3.1 Qualitätsanforderungen an Software-Produkte

„*Die Bedeutung der frühzeitigen und systematischen Erhebung von Anforderungen an Softwareprodukte wird bereits seit Langem erkannt und ist durch entsprechende Projektphasen in Vorgehensmodellen verankert*“ [Geis+06, S. 8]. Neben den projektspezifischen Anforderungen an ein Produkt, gibt es auch eine Reihe weiterer Bedingungen, die jede gute Software erfüllen sollte. Viele Autoren sind sich darüber einig, welche die wichtigsten Qualitätsanforderungen an Software-Produkte sind. In Bezug auf die Struktur des Programmcodes nennen sie die Faktoren Wiederverwendbarkeit, Wartbarkeit, Modularität sowie Portabilität. Die Funktionalität des Systems betreffend sind Korrektheit, Zuverlässigkeit, Integrität, Robustheit und Effizienz die meist genannten Bedingungen. Benutzerfreundlichkeit, Flexibilität und Verständlichkeit gehören zu den anwenderspezifischen Anforderungen [vgl. PaSi94]. All diese Faktoren werden im Folgenden kurz erläutert.

3.1.1 Anforderungen an die Implementierung

Die *Wartbarkeit* einer Software wird durch eine gute Dokumentation sowie einfache Korrigier- und Erweiterbarkeit des Programmcodes gegeben. Durch *Modularität*, also

die Aufteilung des Systems in voneinander unabhängige Bausteine, kann die Erweiterbarkeit vereinfacht werden. *Portabilität* ist die Übertragbarkeit der Software auf andere Umgebungen, also Flexibilität in Bezug auf Hard- und Software sowie das Betriebssystem. *Wiederverwendbarkeit* hingegen ist die Möglichkeit, einzelne Programmteile in andere Software-Produkte einzubauen, ohne dafür große Veränderungen am Code vornehmen zu müssen.

3.1.2 Anforderungen an die Funktionalität

Korrektheit bezieht sich auf die Erfüllung der in der Spezifikation festgelegten Funktionen und produktspezifischen Anforderungen. *Zuverlässigkeit* bedeutet eine geringe Fehlerquote, während *Robustheit* die Ausfallsicherheit bei unerwarteten Ereignissen (wie beispielsweise bei Fehlbedienung oder Fehlfunktionen der Hardware) sicherstellt. *Effizienz* ist die gute Ausnutzung der Systemressourcen und wird gemessen am Laufzeitverhalten und dem Bedarf an Speicherplatz. Ebenso ist die *Integrität* (Verlässlichkeit, Unversehrtheit) der Daten und Funktionen von Bedeutung.

3.1.3 Benutzerspezifische Anforderungen

Die Anforderungen der *Verständlichkeit* und *Benutzerfreundlichkeit* lassen sich als Teile einer guten *Usability* sehen. Diese und eine hohe *Flexibilität* sind zwei weitere wichtige Anforderungen. Im Gegensatz zu den bisher genannten beziehen sich diese beiden nicht nur auf technische Anpassungen sondern betreffen den Nutzer und dessen Interessen direkt. Oft herrscht jedoch eine Kommunikationslücke zwischen Entwicklern und Anwendern. Die Entwickler sprechen eher die Sprache der Rechner und denken auch in gleicher Weise, was dazu führt, dass sie eventuell Bedürfnisse der Anwender vernachlässigen [vgl. CoPi+06, S. 144]. Somit sind vor allem für nutzereinbeziehende Verfahren diese beiden Faktoren von besonderer Bedeutung.

Flexibilität

Obwohl sie das gleiche Produkt nutzen, können verschiedene Anwender äußerst unterschiedliche Wünsche an dieses System haben. Deshalb sollten interaktive Systeme, selbst wenn diese für einen speziellen Anwendungsbereich entwickelt

wurden, flexibel gestaltet sein [vgl. CoPi+06, S. 143]. Bei mancher Software betrifft diese Flexibilität lediglich das Design, andere Systeme bieten auch die Möglichkeit verschiedene Zusatzfunktionen auszuwählen. Doch nicht nur für unterschiedliche Anwender ist die Flexibilität eines Systems von Bedeutung. Der dauerhafte Gebrauch einer Software verändert den Anwender und dessen Nutzungsverhalten [vgl. CoPi+06, S. 143]. Aus diesem Grund sollte ein System auch im späteren Gebrauch noch modifizierbar sein.

Usability

“Usability is concerned with making interfaces consistent, controllable, and predictable, which in turn makes them easy and satisfying to use” [LaPr02, S. 2]. Bereits in Kapitel 2.2.3 wurde beschrieben, wie wichtig die Usability für community-unterstützende Systeme ist. Doch nicht nur bei diesen ist die Gebrauchstauglichkeit von großer Bedeutung. Für jedes Software-Produkt sind die drei Prinzipien nach Shneiderman wichtige Qualitätskriterien: Konsistenz, Kontrolle und Vorhersagbarkeit.

In der Analyse zu Beginn eines Projekts werden diese Nutzungsanforderungen jedoch selten genauer untersucht und *„in den Anforderungsdokumenten oft mit geringer Priorität behandelt oder pauschal als sog. `ergonomische Standardanforderungen` ausgewiesen“* [Geis+06, S. 8]. Um diesem Mangel entgegenzuwirken, entwickelte die Deutsche Akkreditierungsstelle Technik GmbH (DATech) 2006 ein Verfahren und ein zugehöriges Handbuch zur Prüfung des Usability-Engineering-Prozesses. Produkt- und Prozessqualität unterscheiden sich darin, dass ein Prozess nie besser sein kann als die durch ihn erzeugten Produkte, jedoch ein gutes Produkt auch zufällig aus einem nicht optimalen Prozess entstehen kann [vgl. Geis+06, S. 9].

Anhand des DATech-Prüfhandbuchs lassen sich Projekte in unterschiedliche Reifegrade einteilen. Dabei zeichnet sich ein Usability-Entwurfsprozess, der das Anfangsstadium überschritten hat, hauptsächlich dadurch aus, *„dass die Produktidee anhand eines zu entwickelnden Nutzungskonzepts (...) zusammen mit Anwendern und Benutzern verfeinert wird, um die Validität der Anforderungen an die Nutzungsqualität so früh wie möglich zu sichern“* [Geis+06, S. 23]. Ein Anforderungs-

Entwickler hat dabei die Aufgabe, den gewünschten Interaktionsablauf zwischen Anwender und System zu entwerfen und zu beschreiben. Ein Usability-Entwickler hingegen realisiert einen Prototypen, der dieses Interaktions-Schema für die Anwender veranschaulicht [vgl. Geis+06, S. 17].

Als graphische Veranschaulichung eines konsequent umgesetzten Usability-Qualitätsmanagements dient der *Design-Use-Cycle* [vgl. Geis+06, S. 25-28]. Dieser ist eine Erweiterung des 1989 von Floyd et al. entwickelten STEPS-Modells und wird in Kapitel 3.2.7 genauer erläutert. Ziel eines optimalen Usability-Engineering ist es, dass *„eine zyklisch geplante Verbesserung von Produkt und Umfeld (...) zur periodischen Verbesserung der Nutzungsqualität und der Nutzungsbedingungen (führt)“* [Geis+06, S. 25].

3.2 Modelle der Software-Entwicklung

Ein Vorgehensmodell legt fest, welche einzelnen Schritte eines Projekts in welcher Reihenfolge durchgeführt werden. Seit den 1970er Jahren haben sich in der Software-Entwicklung viele verschiedene solcher Modelle entwickelt, die Bekanntesten werden hier vorgestellt.

3.2.1 Wasserfallmodell

Das Wasserfallmodell [vgl. Royc70] wird auch als klassisches Phasenmodell bezeichnet. Der Entwicklungsprozess wird in verschiedene Phasen aufgeteilt, die nacheinander abgearbeitet werden. Jede Phase liefert ein Zwischen- bzw. Endergebnis, das in die jeweils nächste Phase mit eingeht.

Vorteile dieses Modells sind die klare Strukturierung des Prozesses und die ebenfalls klare Definition von Zwischen- und Endprodukten, die als Meilensteine zur Kontrolle des Projektfortschritts verwendet werden können. Die größte Schwäche des Wasserfallmodells ist der strikt lineare Ablauf, der keinen Rücksprung in frühere Phasen erlaubt. Dies wäre beispielsweise zur Fehlerbehandlung oder bei Veränderungen der Produkthanforderungen notwendig. Somit ist das Projekt nur auf einfache Projekte anwendbar. Des Weiteren sind Tests erst am Ende der

Produktentwicklung möglich, wodurch Fehlerkorrekturen sehr aufwendig werden können.

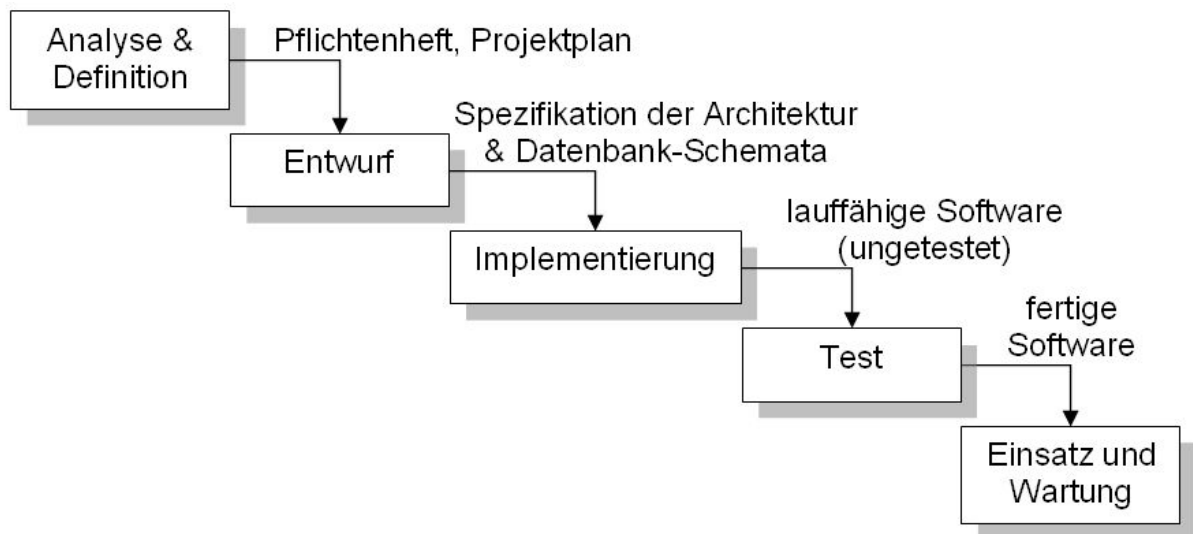


Abbildung 2: Wasserfallmodell nach Royce [vgl. Royc70]

3.2.2 Iteratives Phasenmodell

Das iterative Phasenmodell [vgl. TsSt+97] erweitert das Wasserfallmodell um Rücksprünge in frühere Phasen, wodurch jedoch die lineare Abfolge gestört wird. Da sowohl Zeitpunkt als auch Umfang der Rücksprünge im Voraus nicht bekannt sind, bevorzugen andere Modelle eine parallele Bearbeitung der einzelnen Schritte für verschiedene Module des Endproduktes.

3.2.3 Rapid Prototyping

Das Problem der hohen Fehlerbehebungskosten kann durch Rapid Prototyping [vgl. ThWi92] gelöst werden. Schon früh im Entwicklungsprozess wird hierbei eine lauffähige Testversion, der Prototyp des angestrebten Endproduktes, erzeugt. Dieser kann nun immer wieder erweitert und auf Fehler geprüft werden. Vor allem ist dies sinnvoll, wenn die Anforderungen an ein Produkt anfangs noch unklar sind beziehungsweise sich im weiteren Projektverlauf ändern.

3.2.4 Spiralmodell

Das Spiralmodell [vgl. Boeh88] unterteilt den Entwicklungsprozess nicht in einzelne Phasen sondern sieht ihn als eine Reihe von Zyklen, wobei jeder Durchgang aus vier

Schritten besteht. Zuerst werden bei der Problemanalyse Rahmenbedingungen, Ziele und Alternativen bestimmt, danach bei der Risikoanalyse die verschiedenen Alternativen verglichen und bewertet. Nach der Entscheidung für eine Vorgehensweise wird ein Zwischenprodukt erstellt. Abschließend findet die Planung des nächsten Zyklus statt. Am Ende jeder Wiederholung wird anhand eines Reviews der Projektfortschritt gemessen und entschieden, ob das Projekt weiter fortgesetzt oder abgebrochen wird.

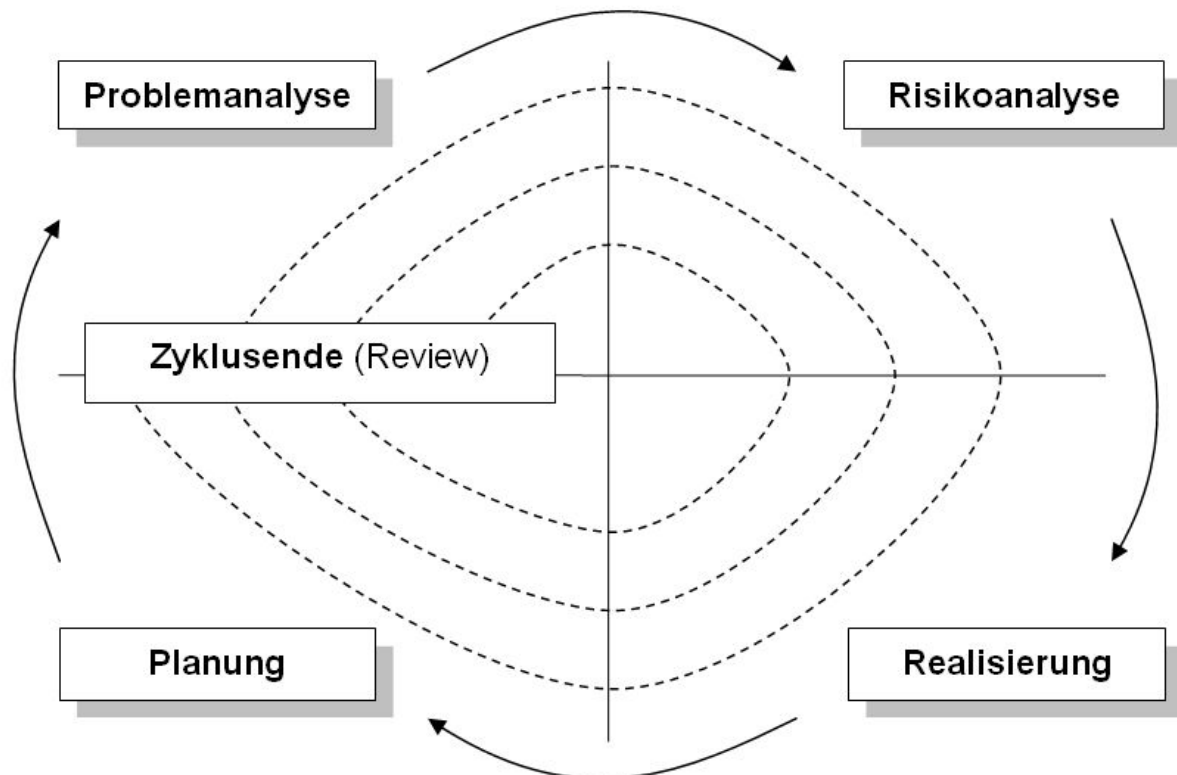


Abbildung 3: Spiralmodell nach Boehm [vgl. Boeh88]

3.2.5 V-Modell

Das V-Modell ist ein vom Bundesministerium für Verteidigung entwickelter Standard, der neben dem Vorgehensmodell auch die Zuordnung von Methoden sowie funktionale Werkzeuganforderungen umfasst [vgl. Pelk00]. Seit dem 4. November 2004 gilt das *V-Modell XT* verbindlich für alle neu zu entwickelnden Softwaresysteme der Bundesverwaltungsbehörden [vgl. Bund04]. Seine Vorgänger waren das ursprüngliche Modell von 1993 und die überarbeitete Version von 1997. Im Gegensatz zu den bisher vorgestellten Modellen, steht bei diesem die zeitliche Abfolge der einzelnen Tätigkeiten nicht im Vordergrund der Betrachtung. Vielmehr

sind eine klare Strukturierung und gute Dokumentation von Bedeutung, um Tests und Qualitätssicherung zu vereinfachen [vgl. Ploe01].

3.2.6 STEPS-Modell

Die Abkürzung *STEPS* steht für Softwaretechnik für evolutionäre partizipative Systemgestaltung. 1989 entwickelten Floyd et al. dieses Modell, das vom zyklischen Ablauf her stark dem Spiralmodell ähnelt. Beim STEPS-Modell liegt jedoch der Schwerpunkt auf der Zusammenarbeit von Entwicklern und Anwendern sowie der Koordination ihrer Aufgaben. Der Entwicklungsprozess wird als eine Reihe von Zyklen gesehen, die aufgrund geänderter Anforderungen der Anwender immer neue Systemversionen hervorbringen [vgl. FIRE+89]. Das STEPS-Modell kann somit als eine Kombination aus dem Spiralmodell und evolutionärem Prototyping verstanden werden.

3.2.7 Design-Use-Cycle

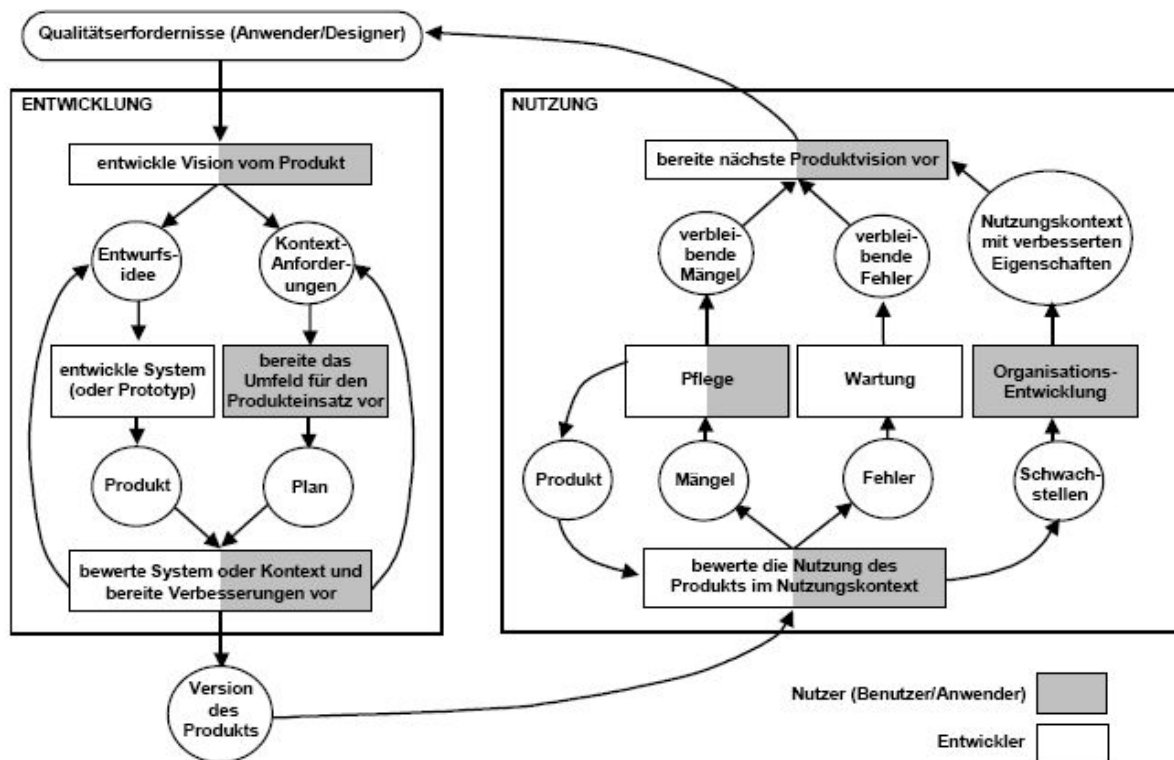


Abbildung 4: Design-Use-Cycle nach Dzida [Geis+06, S. 26]

Der 1997 von Dzida entwickelte Design-Use-Cycle [vgl. Dzid97] ist eine Erweiterung des STEPS-Modells und gibt eine konsequente Umsetzung des Usability-Qualitätsmanagements wieder. Der Entwicklungsprozess wird in die beiden Phasen der Gestaltung und Nutzung unterteilt. Dies soll verdeutlichen, dass sowohl die Software selbst als auch ihr Nutzungskontext sich gegenseitig beeinflussen und weiterentwickeln. Beide Prozessphasen beziehen sowohl die Entwickler als auch die Anwender ein [vgl. Jann06, S. 17-19].

3.3 Design-Konzepte mit Einbeziehung der Nutzer

Eine Ausrichtung auf und Einbeziehung von Endnutzern findet im Software-Design schon seit Langem statt. Bereits seit den 1970er Jahren vertreten Forscher die Meinung, Produkte sollen auf menschlichen Bedürfnissen und Wünschen basieren, nicht auf der zugrunde liegenden Technologie [vgl. Ford94, Levi98, AIEd+01]. Verschiedene Ansätze betrachten den Anwender dabei jedoch aus sehr unterschiedlichen Perspektiven.

3.3.1 User Centered Design

Beim *User Centered Design* (UCD) wird die Umgebung, in der das zu entwickelnde System betrieben werden soll, in Hinsicht auf Eigenschaften sowohl der Nutzer als auch der Organisation, sowie auf Ziele, Aufgaben und den Prozesskontext analysiert. Die Endanwender dienen als Informanten, sind allerdings keine Mitglieder des Design-Teams [vgl. IrEr+04]. In Anlehnung an [PrRo+02] definieren Wania et al. UCD als *“an approach in which users and their goals, not just technology, are the driving force behind the development of a product”* [WaAt+06, S. 91].

3.3.2 Participatory Design

„PD implies that users of computer applications should take part in the decision that affect the system and the way it is used and designed“ [Hans02, S. 1]. Das *Participatory Design* bietet unterschiedliche Methoden, um Wissen und Wertschätzungen der Nutzer direkt in den Designprozess einzubinden und die Anwender selbst aktiv an diesem Vorgang teilhaben zu lassen. Wie Irestig et al. in [IrEr+04] erforschen, ist der Fokus beim PD stärker auf kollektiven Gebrauch und

soziale Prozesse gerichtet, beim UCD jedoch auf die Verbesserung der individuellen Nutzung. Erste partizipative Ansätze gab es bereits in den 1970er Jahren, viele Unternehmen haben seitdem die Vorteile erkannt, die ihnen eine frühzeitige Integration der zukünftigen Anwender bietet.

3.3.3 Open Innovation

Der Begriff *Open Innovation* (oder synonym *User Innovation*) stammt aus der Betriebswirtschaftslehre und steht für die freiwillige, aktive Einflussnahme von Kunden im Innovationsprozess. Reichwald und Piller definieren Open Innovation als „eine interaktive Wertschöpfung im Innovationsprozess, indem ein Herstellerunternehmen mit ausgewählten Kunden bzw. Nutzern gemeinschaftlich Innovationen generiert“ [RePi06, S. 96]. Die Zusammenarbeit erfolgt dabei in den Phasen von der Ideengenerierung, über die Entwicklung erster Lösungskonzepte und die Anfertigung von Prototypen bis hin zur Verbreitung der Innovation. Das Konzept der Open Innovation scheint somit eine noch stärkere Einbeziehung der Anwender vorzusehen als es bei vielen PD-Ansätzen der Fall ist. Außerdem werden im Gegensatz zum PD stärker die betriebsinternen Abläufe betrachtet. Reichwald und Piller erläutern weiterhin, es sei aufgrund der Möglichkeiten, die das Internet heutzutage biete, die wichtigste Aufgabe der Herstellerunternehmen, „Interaktionsplattformen zu schaffen, die den Inputprozess für den Kunden zum Erlebnis werden lassen“ [RePi06, S. 5].

3.3.4 User Driven Development

Ein in der Forschung relativ neuer Begriff ist das *User Driven Development* (UDD), welches dem Bereich der Wirtschaftsinformatik zugeordnet wird. Eine allgemein anerkannte Definition des UDD existiert noch nicht, oft wird es synonym mit der User Innovation gebraucht.

De Moor und van der Rijst entwickelten 1996 *RENISYS*⁴, einen ersten als UDD bezeichneten Ansatz für die Entwicklung von Netzwerk-Informationssystemen [vgl. MoRi96]. Dabei konzentrierten sie sich vor allem auf Netzwerke im Bereich der

⁴ RENISYS = *Research Network Information System Specification Method*

Forschung, bei denen sie die Anwender als äußerst wichtige Beitragende für Analyse und Design der zugrundeliegenden Informationssysteme sahen. De Moor und van der Rijst trennten strikt die nutzergetriebene System-Spezifikation von der nutzerunabhängigen Implementierung. RENISYS sollte sowohl nutzergetrieben als auch kontextsensitiv sein und von Sprachbarrieren abstrahieren, um für Nutzer sowie Entwickler verständlich zu sein. Das Ziel des RENISYS-Projekts war es somit, eine Methode zu liefern, die benutzerdefinierte Anforderungen effizient auf Spezifikationen für ein Informationssystem abbilden sollte [vgl. MoRi96, S. 9].

Christina Hansson hingegen beschreibt UDD in [Hans02] als eine erweiterte Form des PD durch Kombination mit einem guten Customer-Relationship-Management. Dabei nehmen die Nutzer zwar nicht aktiv an Designprozessen und -entscheidungen teil, jedoch baut der Entwicklungsprozess auf einem ständigen Feedback-Strom der Anwender auf. Die Rückmeldungen der Nutzer werden während des gesamten Lebenszyklus eines Systems benötigt und immer wieder als Grundlagen für Updates oder neue Systemversionen verwendet. Somit kann das System sehr zeitnah an sich verändernde Arbeitsmethoden und -situationen angepasst werden [vgl. Hans02, S. 18, in Anlehnung an KeBI98].

3.3.5 Community Driven Development

Aufbauend auf dem Begriff des UDD wird das im Rahmen dieser Arbeit verwendete Konzept als *Community Driven Development* (CDD) bezeichnet. Die Einbeziehung einer Nutzer-Community sowie deren Möglichkeit zur Diskussion und gemeinsamen Ideenfindung und Entschlussfassung stehen somit im Vordergrund der Betrachtung. Im Gegensatz zum UDD spricht das CDD dem Nutzer bedeutend mehr Mitwirkungs- und Entscheidungskraft zu. Bezogen auf die im Rahmen dieser Arbeit durchgeführte Fallstudie, erläuterte der Leiter des betreffenden Geschäftsbereichs, „*dass die gesamten Anregungen - als sprich das WAS, die Funktionen, aber auch das WIE, also wie diese Funktionen umgesetzt werden - nicht mehr (vom Unternehmen) vorgegeben werden und dann getestet wird, ob es funktioniert, sondern dass die User selber (...) den Input geben, wie sie denn die ideale Entwicklung sehen, und (das Unternehmen) quasi zum Dienstleister für die*

Anwender (wird)“ [Anhang – Interview 06]. Diese Begriffsbestimmung gilt für die Diplomarbeit als Grundlage.

3.4 Vorteile und Probleme der Nutzerpartizipation

Macdonald et al. beschreiben in [MaPe+04] die Vorzüge der Nutzerpartizipation aus verschiedenen Sichtweisen. Geschäftsführer achten heute mehr denn je auf Fokusgruppen, Kundenbefragungen, Markttrendanalysen und auch ethnographische Studien. Gefordert sind weniger neue Anwendungen sondern Verbesserungen bestehender Lösungen in Bezug auf die Usability. Gerade an diesem Punkt ist eine Einbindung der Endnutzer nötig. Viele Anwender sind gerne dazu bereit, am Software-Design teilzunehmen. In einigen Fällen werden ihnen für ihre Mitwirkung Vergünstigungen angeboten. Jedoch werden sie nicht nur durch finanzielle Anreize zur Partizipation angeregt. Laut Füller et al. sind der Spaßfaktor und die Hoffnung auf ein verbessertes, den eigenen Wünschen entsprechendes Produkt viel bedeutender für die Motivation der Anwender [vgl. FÜBa+06, S. 68]. Weitere Faktoren seien Neugierde, Lernbegierde, die Möglichkeit frühzeitig exklusive Informationen über Innovationen zu erhalten, persönliches Interesse am Thema sowie Anerkennung durch andere [vgl. FÜBa+06, S. 63]. Die Nutzer fühlen sich anerkannt und beachtet, bauen eine stärkere Bindung zu Anbieter und Produkt auf.

Ein wichtiger Aspekt für nutzereinbeziehende Designprozesse ist die **Lead User Theorie** [vgl. FÜBa+06, FrHi+05]. Sie beschäftigt sich mit der Fragestellung, welche Anwender geeignet sind, in Design- oder Entwicklungsprozesse einbezogen zu werden. Aufbauend auf mehreren Studien von Eric von Hippel [vgl. FrHi+05, Hipp86, Hippel] lassen sich Lead User als Nutzer definieren, die zwei Eigenschaften erfüllen. Zum Einen sollten sie sich so intensiv mit dem jeweiligen Produkt oder Markt befassen, dass sie frühzeitig auf neue Bedürfnisse aufmerksam werden und somit neue Trends und Notwendigkeiten erkennen, die später für viele Anwender von Bedeutung sein könnten. Zum Anderen sollten sie sich selbst einen hohen Nutzen von der Problemlösung versprechen und dadurch ein verstärktes Interesse an der Beteiligung haben [vgl. Hipp86, S. 796; FrHi+05, S. 3]. Von Hippel teilt den Prozess zur Einbeziehung der Lead User in vier Phasen. Nach der Identifikation eines wichtigen technischen oder Markttrends, steht als zweiter Schritt die Identifizierung

der Lead User anhand ihrer Erfahrungen und Bedürfnisse an. Anschließend werden zusammen mit diesen Personen durch Analyse ihres Bedarfs innovative Produktkonzepte erstellt, um diese zuletzt auf einen größeren Markt zu projizieren [vgl. Hipp86, S. 797].

Um eine sinnvolle Interaktion zwischen Unternehmen und Kunden zu ermöglichen, müssen diverse Voraussetzungen erfüllt werden. Neben den Anwendern müssen auch die Mitarbeiter des Unternehmens zur Zusammenarbeit bereit sein und sich darauf einstellen. Reichwald und Piller bezeichnen dies als **Interaktionskompetenz**, die sich in interaktionsfördernde Kommunikations-, Anreiz- und Ablaufstrukturen unterteilen lässt [vgl. RePi06, S. 81-91]. Oft herrscht bei Entwicklern das *Not Invented Here (NIH) Syndrom*, ein Widerstand gegen die Einbringung externen Wissens. Um dieses Problem zu überwinden, sollen sogenannte Gatekeeper das Entwicklungsteam mit externen Quellen verbinden und dabei unbrauchbare Informationen herausfiltern. Microsoft setzt beispielsweise Gatekeeper ein, die bereits seit vielen Jahren im Unternehmen tätig sind und somit das Vertrauen der Entwickler genießen. Diese Personen sind auf unterschiedlichen Ebenen für die Zusammenarbeit mit den Lead Usern verantwortlich [vgl. RePi06, S. 89]. Eine weitere Möglichkeit, die Motivation der Mitarbeiter zu erhöhen, ist eine Öffnung der Unternehmensstruktur, so dass Entscheidungen jeweils zu der Unternehmensebene delegiert werden, in der sie anstehen. So genießen die Mitarbeiter mehr Vertrauen, steigende Verantwortung wirkt meist motivierend. Weiterhin können Instrumente genutzt werden, die den Austausch von Wissen unterstützen, wie beispielsweise Job Rotation oder interfunktionale Gruppen. Die Kommunikation zwischen Unternehmen und Kunden sollte ein kontinuierlicher Dialog sein, wozu eine unmittelbare Erreichbarkeit- und Interaktionsmöglichkeit nötig ist. Durch eine Vernetzung der Kommunikationsstruktur kann eine größere Reichweite ebenso gewährleistet werden wie die Möglichkeit, Beiträge aufeinander aufzubauen. Die Integration der Kundenbeiträge in die Wertschöpfungskette kann auf drei Arten ablaufen. Eine automatische Integration über die Kommunikationsplattform ist vor allem bei kleinen Beiträgen denkbar. Dieses Verfahren wird jedoch hauptsächlich im Bereich der Mass Customization angewandt. Da bei der Open Innovation prinzipiell neue Ideen entwickelt werden, ist eine automatische Anpassung meist nicht möglich. Somit muss

die Integration von Mitarbeitern oder Kunden des Unternehmens vorgenommen werden. Ein bekanntes Beispiel für die Peer Production, also die Einführung der Beiträge durch die Anwender selbst, ist die Online-Enzyklopädie Wikipedia⁵. Bei dieser bringen die Nutzer nicht nur eigene Beiträge in das System ein, sondern ergänzen und verbessern auch die Beiträge anderer Anwender. Im Gegensatz dazu ist der Software-Hersteller Stata Corp ein Beispiel für die unternehmensinterne Abwicklung [vgl. RePi06, S. 88]. Stata Corps statistische Software erlaubt dem Anwender die einfache Programmierung neuer Tests. So teilt sich die Software in einen kommerziellen, vom Unternehmen entwickelten Teil, sowie einen offenen Teil, der von den Anwendern weiterentwickelt wird. Stata Corp bietet seinen Kunden dazu eine Entwicklungsumgebung sowie ein Online-Forum, in dem die Anwender ihre Ideen austauschen können. Da jedoch nicht jeder Kunde die Kenntnisse hat, um neue Tests selbst einzubauen, wählt das Entwickler-Team des Herstellers regelmäßig Nutzer-Ideen aus, die in die kostenpflichtige Software-Version integriert werden. Die Einbindung ist somit weiterhin vom Unternehmen bestimmt, die Anwender dienen lediglich als Ideengeber.

3.5 Techniken zur Nutzereinbindung

Noch vor einigen Jahren galten Beobachtungen und Interviews als die einzigen Methoden, um Nutzerinformationen zu sammeln [vgl. JäMa03]. Vor allem im letzten Jahrzehnt wurden jedoch viele neue Techniken entwickelt und erforscht. Um einen Überblick über den aktuellen Stand der Forschung zu geben, wird im Folgenden eine Auswahl unterschiedlicher Techniken zur Nutzerintegration in den Designprozess vorgestellt. Meist werden mehrere dieser Methoden in Kombination angewendet.

Muller et al. klassifizieren partizipative Techniken nach zwei Dimensionen. Zum Einen wird das Ausmaß der direkten Einbeziehung des Nutzers in den Designprozess betrachtet, also ob er aktiv gestaltend eingreift oder beispielsweise lediglich beobachtet wird; zum Anderen wird danach unterschieden, in welcher Phase des Entwicklungsprozesses diese Aktivität stattfindet [vgl. MuWi+92]. Abbildung 5 zeigt die Einteilung der wichtigsten nutzereinbeziehenden Techniken in

⁵ <http://de.wikipedia.org>

einer solchen Klassifikationsmatrix.



Abbildung 5: Klassifikationsmatrix von Participatory Design Techniken nach Grad der Nutzeraktivität im Design und nach Beginn der Partizipation

3.5.1 Beobachtungen und Interviews

Diese beiden Methoden werden meist kombiniert angewendet. Nutzer werden in ihrem Arbeitskontext von Designern observiert und anschließend in Einzelgesprächen befragt. Vorteil der Beobachtungen ist, dass diese nicht störend in die Arbeit einwirken, jedoch sind sie oft nicht selbsterklärend und es wird nur ein geringer Anteil der möglichen Anwendungsfälle betrachtet. Deshalb ist die Kombination mit Interviews, bei denen offene Fragen geklärt und weitere Gedanken der Anwender eingebracht werden können, sinnvoll. Interviews können formell oder informell sein. Formelle Interviews sind vorbereitet und finden meist abgesondert vom Arbeitsumfeld statt. Informelle Interviews hingegen beantworten direkt im Arbeitskontext auftretende Fragen, sind somit situationsgebunden und nicht vorbereitet [vgl. Hans02, S. 4].

3.5.2 Usability-Untersuchungen

Eine Usability-Untersuchung dient der Feststellung von Problemen der Gebrauchstauglichkeit eines Systems. Diese werden identifiziert und in Redesign-Prozessen verbessert. Man unterscheidet zwischen heuristischer Evaluation und Nutzertests. Bei der heuristischen Evaluation untersucht eine geringe Zahl von Experten die Benutzerschnittstelle des Systems und bewertet anhand bestehender Heuristiken unterschiedliche Kategorien [vgl. AlBa+93]. Im Gegensatz dazu werden bei Nutzertests die Anwender selbst einbezogen. Begleitet von einem Experten erforschen sie das System entweder frei oder führen ihnen vorgebrachte Aufgaben durch [vgl. BrKu05]. Meist bringen diese Vorgänge jedoch nur wenige Informationen ein.

3.5.3 Probes

Beim Probing erhalten Endanwender unterschiedliche Werkzeuge, die ihnen helfen, ihre Erfahrungen und Gedanken bezüglich ihrer Umgebung und Aktionen zu dokumentieren, reflektieren und auszudrücken [vgl. GaDu99, HuMa04]. Ein großer Vorteil ist es, dass der Nutzer selbst bestimmen kann, wann und wie er dies vornimmt. Des Weiteren können gut gewählte Werkzeuge die Kreativität des Anwenders weiter anregen. Je nach Anwendungsfall werden sehr unterschiedliche Probes genutzt, beispielsweise Fotokameras, Aufnahmegeräte und Tagebücher oder auch Land- und Postkarten [vgl. GaDu99, HuMa04]. Ein Problem bei dieser Technik ist jedoch, dass die Probes für Außenstehende gegebenenfalls nur schwer zu verstehen und auszuwerten sind. Durch die Kombination von Probing mit anschließenden Interviews kann dies erleichtert werden.

3.5.4 Workshops, Gruppendiskussionen

In Workshops diskutiert eine kleine Gruppe von Nutzern unter der Leitung eines Moderators. Ziel dabei ist es, Informationen über die Meinungen und Wünsche der Anwender zu sammeln und daraus Strategien abzuleiten [vgl. BrKu05]. Vorteile sind das direkte Zusammentreffen und der Austausch mit den Nutzern, jedoch sind Zeit- und Organisationsaufwand meist groß und es kann zu Verständnisproblemen zwischen Anwendern und Entwicklern kommen.

3.5.5 Innovation Play

Beim Innovation Play stellen die Nutzer in Rollenspielen unterschiedliche Anwendungsfälle nach. Dies hilft einerseits den beobachtenden Designern, Gründe für das Nutzerverhalten besser zu erkennen, andererseits dient es als Inspiration für die Anwender [vgl. JoGr06]. Allerdings können immer nur wenige ausgewählte Anwendungsfälle nachgespielt werden. ExperienceLabs sind an die gewohnte Nutzungssituation angelehnte Test-Umgebungen, die dazu dienen sollen, dass sich die beobachteten Teilnehmer wohl fühlen und natürlich verhalten. Um die anschließende Auswertung zu vereinfachen, sind diese Räume mit Untersuchungsinstrumenten wie beispielsweise Kameras ausgestattet [vgl. Hoon07].

3.5.6 Rapid Prototyping

Ein Prototyp des neuen Systems wird schrittweise weiterentwickelt, so dass schon während der Designphase ein Einblick in die beabsichtigte Funktionalität gegeben und zu neuen Ideen inspiriert wird [vgl. AlBa93]. Auch einbezogene Nutzer können somit immer neue Testobjekte erhalten, jedoch ist der Entwicklungsaufwand relativ hoch. Nielsen unterscheidet zwischen horizontalen und vertikalen Prototypen. Horizontale Prototypen stellen eine Vorausschau auf die Benutzerschnittstelle dar und dienen zu Usability-Tests. Vertikale Prototypen hingegen sind kleine Teile des Gesamtsystems, die zum Test komplexer Funktionen zu gebrauchen sind [vgl. BrKu05, S. 162 in Anlehnung an Niel94].

3.5.7 Participatory Programming

Nutzer nehmen beim Participatory Programming nicht nur am Designprozess teil, sondern wirken auch bei der Implementierung des Systems aktiv mit [vgl. LeMa04]. Somit können sie komplett einbezogen werden, wozu allerdings Programmierkenntnisse vorausgesetzt sind.

4 Einbeziehung einer Nutzer-Community in den Designprozess

In der Software-Entwicklung führt die Beteiligung von Nutzer-Communities bei Open Source Projekten oft zu guten und innovativen Ergebnissen [vgl. RePi06]; im Closed Source Bereich wurde eine solche Einbeziehung ins Software-Design allerdings bisher kaum vorgenommen. Jedoch gab es in verschiedensten anderen Branchen unterschiedliche Entwicklungsansätze, bei denen Nutzer-Communities in Planung und Entwicklung einbezogen wurden. *„Nicht zuletzt begünstigt durch neue Möglichkeiten der Informations- und Kommunikationstechnologien (...) und des Internets gewinnt die interaktive Wertschöpfung auch in vielen Konsumgütermärkten an Bedeutung“* [RePi06]. Einige dieser Anwendungsfälle werden im Folgenden vorgestellt und ihre Forschungslücken ermittelt.

4.1 Fallbeispiel Kite Surfing – Lead User Theorie

In einer Untersuchung der *MIT Sloan School of Management* wurden anhand einer Fallstudie die Basisannahmen der 1986 von Eric von Hippel begründeten Lead User Theorie⁶ untersucht und neue Erkenntnisse in Anlehnung an diese Ergebnisse ermittelt [vgl. FrHi+05]. Die Lead User Theorie beschreibt die Eigenschaften, die Anwender zur Entwicklung innovativer und für den Massenmarkt tauglicher Ideen qualifizieren. Lead User befassen sich intensiv mit dem Produkt, sind der Entwicklung des Marktes somit ein Stück voraus und erkennen neue Trends frühzeitig. Nach von Hippel führt dies dazu, dass sie Ideen liefern, die auch für kommerzielle Zwecke attraktiv erscheinen. Die zweite einen Lead User qualifizierende Eigenschaft ist, dass der Anwender sich selbst einen hohen Nutzen durch seine Beteiligung verspricht und somit viele Ideen liefert. Diese beiden Faktoren sind voneinander unabhängig, was auch durch die Fallstudie von Franke, von Hippel und Schreier bestätigt wurde. Ebenso bewies die Studie die Bedeutung der beiden Faktoren. Eine neue Erkenntnis war, dass die Fähigkeit, Trends frühzeitig erkennen zu können, eine positive Wirkung sowohl auf die Attraktivität als auch die

⁶ Siehe Kapitel 3.4

Häufigkeit der innovativen Ideen hat. Von Hippel hatte in seiner ursprünglichen Theorie lediglich den Einfluss auf die Innovationsattraktivität erkannt. Des Weiteren erkannten Franke et al. einen weiteren Faktor, der sich positiv auf die Anzahl an Ideen auswirkt. Sie bezeichnen ihn als das Vorhandensein von Ressourcen, womit sowohl die technische Expertise der Nutzer als auch ihre Verbindung zu Communities, die bei der Ideenfindung eine Hilfestellung geben können, gemeint sind.

Franke et al. wählten für ihre Studie das Untersuchungsfeld Kite Surfing, da vor allem im Sportbereich viele Nutzer ihre gekauften Produkte weiter modifizieren; unter den befragten Anwendern betraf dies über 30 Prozent. Die Anwender wurden über Online-Communities zum Thema Kite Surfing kontaktiert, indem über die Internetseiten und Newsletter der Communities Fragebögen herausgegeben wurden. Insgesamt beteiligten sich 456 Anwender an der Studie, dies seien in etwa 30 Prozent der angeschriebenen Nutzer. Mittels der Fragebögen wurde zunächst ermittelt, wer die Lead User Charakteristika erfüllte. So wurde die Positionierung an der Spitze der Marktentwicklung anhand der Ergebnisse der Sportler bei Wettbewerben gemessen, die Erwartungshaltung an zwölf unterschiedlichen Messgrößen und das Vorhandensein von Ressourcen anhand von 18 Werten, die das Vorkommen von technischer Expertise und communitybasierten Ressourcen genauer erfragten. 30,9 Prozent der Befragten gaben an, bereits eigene Verbesserungsideen für das Kite Surfing gehabt zu haben. Ihre ausführlich beschriebenen Ideen wurden sowohl durch andere Anwender als auch sechs Experten untersucht, um die kommerzielle Attraktivität der Innovationen zu beurteilen. 88 Ideen wurden von den Experten als innovativ bewertet, somit galten 19,5 Prozent aller Befragten als Innovatoren.

Die Fallstudie von Franke et al. diente der Überprüfung der Lead User Theorie. Eine weitere Einbeziehung nach der Identifikation der Lead User fand jedoch nicht statt. Ihre Partizipation wurde lediglich durch die Fragebögen und einige wenige Interviews vorgenommen, eine aktivere Einflussnahme wurde nicht gefördert. Im neuen Konzept soll die Nutzer-Community über den kompletten Entwicklungsprozess

hinweg einbezogen werden. Dabei sollen die Anwender ein bedeutendes Mitsprache- und Bestimmungsrecht haben.

4.2 Studien nach Reichwald und Piller – interaktive Wertschöpfung

Als interaktive Wertschöpfung bezeichnen Reichwald und Piller die Arbeitsteilung zwischen einem Herstellerunternehmen und dessen Kunden, wenn die Kunden Produkte und Dienstleistungen als aktive Wertschöpfer mitgestalten, gegebenenfalls sogar deren Entwicklung bestimmen oder selbst übernehmen [vgl. RePi06]. Dabei unterscheiden die Forscher zwischen den Begriffen *Open Innovation* und *Mass Customization*. Während Open Innovation sich auf Aktivitäten im Innovationsprozess bezieht und somit die Entwicklung neuer Produkte für einen großen Abnehmerkreis erzielen soll, bezieht sich die Mass Customization auf Aktivitäten der Produktion und somit eine Individualisierung der Produkte für einzelne Kunden.

Die Mass Customization wird meist durch Konfigurationstools erreicht, die es den Anwendern ermöglichen, verschiedene Anpassungen vorzunehmen, die das Produkt ihren Wünschen annähern. Reichwald und Piller stellen in ihrem Buch [RePi06] verschiedene Ansätze zu diesem Prinzip vor. Im Rahmen dieser Arbeit interessiert jedoch stärker der noch weniger erforschte Bereich der Open Innovation.

Wie bereits in Kapitel 3.3.3 erläutert, kann die Beteiligung der Kunden bei der Open Innovation die Phasen der Ideengenerierung, Konzeptentwicklung, die Erstellung von Prototypen, Produkttests sowie letztlich die Einführung des Produktes am Markt betreffen. Dabei steht vor allem die freiwillige Arbeit der Anwender im Vordergrund. Sie müssen jedoch nicht als autonome Nutzer ohne Interaktion mit dem Unternehmen neue Ideen komplett entwickeln, wie dies beispielsweise bei von Hippels Kite Surfing Studie der Fall war. Reichwald und Piller bevorzugen einen expliziten Kooperations- und Interaktionsprozess zwischen Kunden und Unternehmen, der zusätzlich durch unterschiedliche Toolkits unterstützt werden kann [vgl. RePi06, S. 47].

Umstritten ist die Frage nach der Motivation der Anwender. Viele Kunden seien dazu bereit, ihre Ideen einzubringen ohne eine Gegenleistung zu verlangen, was als **Free**

Revealing bezeichnet wird. Gründe dafür seien vor allem die Vorteile für die eigene Nutzung. So entwickeln zwar einige Anwender Prototypen für den eigenen Gebrauch, professionelle Anfertigungen hingegen könnten jedoch eine bessere Qualität bieten. Des Weiteren könnten diese Produkte weiterentwickelt und verbessert werden, was letztlich für alle Anwender von Vorteil sei. Herrsche keine oder nur eine geringe Rivalität im Anwendungsfeld, so seien viele Anwender bereit, ihre Ideen einzubringen, was ihnen zusätzlich eine positive Reputation und oft auch Spaß an der Zusammenarbeit bringe [vgl. RePi06, S. 72-73]. Viele Forscher halten es somit für unnötig, den Anwendern für ihre Beteiligung eine weitere Belohnung zu bieten. Jedoch sind einige Unternehmen trotzdem zu einem solchen Schritt bereit. Es existieren immer mehr **Innovationsnetzwerke**, die Anwendern größere Anreize für ihre Ideen bieten [vgl. RePi06, S. 96-97, 114-116]. In diesen Netzwerken stellen Unternehmen anonym Fragen oder Aufgaben online und geben dazu ein Preisgeld an. Die angemeldeten privaten Problemlöser können ihre Ideen einsenden. Nutzt das Unternehmen eine dieser Lösungen, so erhält der Einsender das Preisgeld, beim Netzwerk *InnoCentive* beispielsweise liegt der Mindestbetrag bei 10.000 US Dollar. InnoCentive gehört mit 30 suchenden Unternehmen und über 70.000 registrierten Teilnehmern zu den größten Innovationsnetzwerken. Manche Firmen wie zum Beispiel *Procter & Gamble* gründen jedoch auch eigene Netzwerke für ihre Kunden.

Reichwald und Piller identifizieren vier **Instrumente der Open Innovation**: die Lead User Methode, Toolkits, Innovationswettbewerbe sowie Communities [vgl. RePi06, S. 155; Lech07]. Entsprechend der Lead User Theorie nach von Hippel können innovative Anwender identifiziert und über Innovationsworkshops in die Entwicklung eingebunden werden. Toolkits sollen den Anwendern helfen, ihre eigenen Vorstellungen und Bedürfnisse in neue Konzepte zu übertragen. Dabei sollen diese meist internetgestützten Werkzeuge einen vollständigen Trial-and-Error-Prozess abbilden, so dass die Kunden iterativ verschiedene Varianten und Kombinationen testen und selbständig zu einer Lösung finden können. Innovationswettbewerbe hingegen dienen der Gewinnung innovativer Ideen in frühen Phasen, wobei der Wettbewerb zwischen verschiedenen Nutzern deren Kreativität anregen soll. Auch Reichwald und Piller erkennen den Nutzen, den die Zusammenarbeit einer

Community bieten kann. So könne eine virtuelle Gemeinschaft gemeinsam Ideen generieren und bewerten.

Diese Möglichkeit nutzte auch der Sportartikelhersteller Adidas-Salomon und veranstaltete einen Ideenwettbewerb, um seine Kunden an der Weiterentwicklung eines Konfigurationsmechanismus zu beteiligen, mit dem Schuhe individuell an Passform, Design- und Funktionswünsche der Kunden angepasst werden können. Über ein Online-Tool konnten die Anwender ihre eigenen Ideen einbringen und die Vorschläge anderer Teilnehmer bewerten oder deren Gedanken weiterführen. Parallel zu diesem Prozess füllten die Mitwirkenden Fragebögen aus, mittels derer sie auf ihre Lead User Eigenschaften untersucht wurden. Es stellte sich heraus, dass die kreativsten Ideen von Anwendern stammten, die diese Charakteristika erfüllten. Weiterhin bemerkten die Forscher, dass viele der Ideen, die eine solche Studie hervorbringt, unbrauchbar sind oder sich wiederholen. Eine gute Filterung sei somit dringend notwendig.

Reichwald und Piller sehen die Einwirkung der Kunden in allen Phasen des Innovationsprozesses als besonderen Vorteil der Open Innovation. Die von ihnen vorgestellten Beispiele beziehen sich jedoch meist nur auf die Mitwirkung der Anwender in einem einzelnen Bereich. Es wurde kein Konzept vorgestellt, das eine durchgehende Beteiligung der Nutzer über alle Phasen hinweg unterstützt.

4.3 Fallbeispiel Audi – Community Based Innovation

Füller et al. untersuchten 2006 die theoretischen Möglichkeiten, virtuelle Communities in die Entwicklung neuer Produkte einzubeziehen und prüften ihre Überlegungen anhand einer Fallstudie in Zusammenarbeit mit dem deutschen Automobilhersteller Audi [vgl. FÜBa+06]. Dabei legten sie neben der Lead User Theorie ebenso viel Wert auf die Netnographie bzw. Online-Ethnographie. Diese untersucht die Kommunikation von Mitgliedern großer Online-Communities und liefert somit Informationen über deren Sprachstruktur sowie darüber, wie man die Communities identifiziert, mit ihren Mitgliedern in Kontakt tritt und ihr Verhalten analysiert.

Schwierigkeiten bei der Übertragung dieser Theorien auf den Untersuchungsfall waren, dass die Netnographie lediglich auf Beobachtungen basiert und die Lead User Theorie auf Offline-Nutzer und Industriegüter spezialisiert ist [vgl. FÜBa+06, S. 61]. Als Kombination und Weiterentwicklung der beiden Ansätze entwickelten Füller et al. eine neue Methodik, bei der sie die Nutzer online in ihre Studien einbezogen. Den Prozess dieser *Community Based Innovation* (CBI) unterteilen sie in vier Phasen. In der ersten Phase werden die Merkmale bestimmt, die einen zur Einbeziehung geeigneten Nutzer auszeichnen. Dies geschieht in Abhängigkeit von der jeweiligen Aufgabe. Sind die entscheidenden Kennzeichen ermittelt, so besteht der zweite Schritt in der Identifikation einer Community, in der sich Anwender mit solchen Eigenschaften aufhalten. Die dritte Phase befasst sich mit dem virtuellen Interaktionsdesign, das sowohl an die jeweilige Aufgabe als auch an die einbezogenen Anwender angepasst sein sollte [FÜBa+06, S. 63]. Unter Berücksichtigung der Sprache und Normen der Community sowie der persönlichen Motivation der Anwender kann ein eigenes Konzept entwickelt oder eines der vielen verfügbaren Toolkits dazu eingesetzt werden, die Kommunikation zu unterstützen. Die letzte Phase der CBI besteht in der tatsächlichen Einbeziehung der Anwender von der Kontaktaufnahme bis hin zur Partizipation im Designprozess. Den Nutzern sollte dabei ein Ansprechpartner gegeben werden, der sie über den gesamten Projektverlauf zuverlässig und vertrauensvoll unterstützt. Ein idealer Entwicklungsprozess besteht nach Füller et al. aus drei Stufen: Ideen- und Konzept-Generierung, Design und Konstruktion sowie Tests und Einführung. In diesen drei Stadien kann der Nutzer unterschiedliche Rollen übernehmen – als Ideenquelle, Co-Designer oder möglicher Endanwender [vgl. FÜBa+06, S. 58-59]. Virtuelle Toolkits können den Nutzern die Möglichkeit geben, webbasiert eigene Produktkreationen zu erzeugen und in Simulationen zu testen. Auf diese Weise dienen die Nutzer nicht nur als Ratgeber, sondern können aktiv mitgestalten. „*Members of online communities who are characterized by high product and activity involvement, represent an ideal resource for co-designing products when confronted with those new methods*“ [FÜBa+06, S. 59]. In der abschließenden Testphase können die Community-Mitglieder als mögliche Endanwender ihre Meinung zu den angebotenen Produkten und deren Vermarktungsstrategie angeben.

In Zusammenarbeit mit der Audi AG prüften Füller et al. ihr Konzept an einer dreimonatigen Fallstudie zum Entwurf eines neuen Infotainment Systems. Die zugrundeliegende Community bestand aus allen Nutzern der Audi-Homepage und den Empfängern des zugehörigen Newsletters, insgesamt waren 1662 Kunden an der Studie beteiligt. Die Zusammenarbeit fand über das sogenannte *Virtual Lab* statt, in dem jeder Beteiligte virtuell seine Vorstellungen über Konfigurationen und Dienste des Services sowie seine eigene Ideen einbringen konnte. Des Weiteren wurden Informationen über die Fachkenntnisse und repräsentativen Fähigkeiten der einzelnen Kunden erfragt. Mit Hinblick auf die Lead User Theorie wurden diese Daten analysiert, um die Nutzer in unterschiedliche Gruppen einzuordnen und ihre Angaben somit speziellen Aspekten und Aufgaben zuzuordnen. Positiv hervorzuheben waren vor allem die hohe Beteiligungsbereitschaft sowie das Fazit, dass 80 Prozent der beteiligten Kunden erneut an einem solchen Prozess mitwirken würden. Schwerpunkt in dieser Fallstudie waren die Ermittlung von Kundenwünschen und aufkommenden Trends sowie die Erkenntnis, dass eine virtuelle Plattform wie das Virtual Lab von Anwendern akzeptiert und genutzt wird.

Füller et al. erstellten in ihrer Studie eine Liste der Chancen und Risiken ihrer Technik [vgl. FÜBa+06, S.70]. So ist eines der von ihnen ermittelten Probleme, dass die Meinungen einzelner Kunden eventuell nicht die Interessen des Zielmarktes widerspiegeln. Aus diesem Grund sollte die Gruppe der einbezogenen Anwender nicht zu klein ausfallen, so dass repräsentative Meinungen erfasst werden können. Weiterhin erklärten Füller et al., die Anwender würden sich oft nur an bereits existierenden Lösungen orientieren und somit kaum neue Ideen einbringen. Hätten sie Wünsche, so würde es ihnen außerdem schwer fallen, diese zu formulieren und auf Produktebene zu übertragen. Der Interaktionsprozess, der notwendig ist, um die Nutzer erfolgreich einzubeziehen, erfordert zusätzlichen Aufwand und Ressourcen. Jedoch sollte sich dies in den meisten Fällen durch die Mitwirkung der Anwender und deren Übernahme von Aufgaben, die sonst Entwickler erledigen müssten, ausgleichen. Allerdings merken Füller et al. weiterhin an, dass die Interaktion mit der Community einen störenden Effekt auf die internen Prozesse haben und somit den Fluss des Projektverlaufs hindern könne. Ein weiteres entdecktes Risiko war die Möglichkeit, dass konkurrierende Unternehmen jemanden in den Prozess

einschleusen könnten, der somit an Informationen gelangen könnte, die ansonsten geheim gehalten würden. Zu diesem Zweck empfiehlt es sich, nur Anwender einzubeziehen, die bereit sind, einer Vertraulichkeitserklärung zuzustimmen. Als weiteres rechtliches Problem nennen Füller et al., dass es für das Unternehmen schwierig sei, legale Ansprüche auf die Ideen der Anwender zu erhalten. Auch dies sollte im Voraus schriftlich festgelegt werden. Doch auch die Chancen der Nutzerpartizipation werden von den Forschern hervorgehoben. So sehen sie als einen Vorteil, dass die Fehlerrate bei neuen Produktveröffentlichungen reduziert wird. Dies trifft auf alle Konzepte zu, die eine ausgiebige Testphase beinhalten. Ein weiterer Vorteil sei die Beschleunigung des Innovationsprozesses durch die Mitwirkung der Anwender. Dies ist jedoch ein umstrittener Punkt, die Nutzer können zwar viele neue und innovative Ideen einbringen, jedoch erfordert dies viel Diskussion, wodurch sich die Prozessdauer verlängern könnte. Laut Füller et al. ermöglicht die Kundenorientierung weiterhin eine Verbesserung des Unternehmensimages sowie eine verstärkte Bildung von Kundenbeziehungen und Erhöhung der Kundenbindung. Wird intensiv auf die Anwender eingegangen, so bildet sich ein tieferes Verständnis für ihr Verhalten, was es möglich macht, neue Trends und Veränderungen frühzeitig zu erkennen. Bindet man die Anwender über eine längere Zeit und mehrere Projekte hinweg in die Software-Entwicklung ein, so verstärkt sich die Bindung zwischen Nutzern und Unternehmen. Im Idealfall baut sich eine *Community of Innovation* auf, eine aufeinander eingespielte Gemeinschaft aus Kunden und Entwicklern, in der der Ablauf gemeinsamer Entwicklungsprojekte immer mehr optimiert wird [vgl. FÜBa+06, S. 69].

Zwar untersuchten Füller et al. die Einbindung einer Nutzer-Community in den Designprozess, jedoch stellten sie dabei nicht heraus, worin sich diese von der Mitwirkung einer Gruppe von Anwendern unterscheidet, die sich zuvor unbekannt sind. Zu dieser Fragestellung lieferte auch eine weitreichende Literaturrecherche keine Informationen. Des Weiteren betonten Füller et al., dass unter anderem die Motivation der teilnehmenden Kunden von entscheidender Bedeutung sei und genauer untersucht werden müsse. Somit werden diese Aspekte in unserer Fallstudie genauer untersucht.

4.4 Fallbeispiel Idavall – UDD in der Software-Entwicklung

Christina Hansson begleitete für ihre Master Thesis das schwedische Unternehmen Idavall bei der Entwicklung einer neuen Version des Buchungssystems FRI [Hans02]. Die Firma hält intensiven Kontakt zu ihren Kunden und bindet diese in den Entwicklungsprozess ein. Idavall organisiert etwa acht bis zehn Mal jährlich Meetings an unterschiedlichen Orten des Landes. Bei diesen Treffen begegnen sich Anwender aus der Umgebung und tauschen ihre Erfahrungen ebenso untereinander wie mit Entwicklern aus. Bei den Meetings handelt es sich jedoch nicht um Design-Workshops, gegebenenfalls werden zwar Probleme mit der Software aufgedeckt, dies ist allerdings kein primäres Ziel. Des Weiteren veranstaltet Idavall Produktdemonstrationen zur Gewinnung von Neukunden und Kurse zur Einführung neuer Nutzer in die Softwarebedienung. Neben den bei diesen Begegnungen genannten Ideen und Wünschen der Kunden ist es meist ein telefonisches Feedback der Anwender, das neue Anregungen liefert. Das Konzept des Participatory Design sieht einen festen Entwicklungsprozess vor, zusätzlich sind im Software-Design oft auch Updates zwischen Softwareversionen üblich. Auch Idavall erzeugt jährlich etwa 20 Updates. Dieser kleinere Entwicklungszyklus orientiert sich stark an den Ideen der Nutzer; es werden Produktfehler behoben und gegebenenfalls auch kleinere Zusatzfunktionen auf Kundenwunsch eingebaut.

Bei Idavall nehmen die Nutzer nicht direkt am Designprozess teil, geben jedoch Vorgaben, was ihrem Wunsch nach passieren soll. Meist findet der Kontakt zu den Anwendern telefonisch statt. Die Organisation von Workshops zur stärkeren Einbindung der Nutzer beschreibt Hansson als schwierig, da die Anwender über ganz Skandinavien verteilt leben. Um diesem Problem in der im Rahmen dieser Arbeit begleiteten Fallstudie zu umgehen, wird auf den virtuellen Raum ausgewichen, um einen regelmäßigen Austausch mit den Anwendern zu ermöglichen. Eine weitere Lücke in Hanssons Forschung ist ihre ausschließliche Betrachtung der Entwicklerseite; sie führte keine Gespräche mit den Anwendern.

5 Community Driven Development

Aufbauend auf der untersuchten Literatur und den Forschungslücken der vorgestellten Fallstudien wird im Folgenden ein neues Konzept erarbeitet, über das eine Nutzer-Community in den Software-Designprozess eingebunden werden kann.

5.1 Motivation

Die vorgestellten Fallstudien weisen unterschiedliche Forschungslücken auf. In keinem der untersuchten Projekte wurden die Anwender so weit in den Entwicklungsprozess einbezogen, dass sie den Entwicklern gleichgestellt gewesen wären. Zwar konnten die Nutzer ihre Wünsche äußern und teilweise auch an der Entwicklung teilhaben, sie hatten jedoch keine Entscheidungsgewalt. Außerdem wurde in früheren Untersuchungen überwiegend die Entwicklerseite betrachtet, die Meinungen der Anwender zu nutzereinbeziehenden Konzepten wurden kaum erforscht. Keiner der Forscher konnte die Frage beantworten, inwiefern sich die Einbeziehung einer Nutzer-Community von der Mitwirkung einer Gruppe von Anwendern, die sich zuvor unbekannt sind, unterscheidet. Aufbauend auf dieser Fragestellung und dem Wunsch, eine tatsächlich nutzergesteuerte Entwicklung anzustreben, wurde im Rahmen dieser Arbeit ein neues Konzept des Community Driven Development entwickelt. Dabei sollen die Anwender nicht nur in die anfängliche Planung, sondern über den gesamten Entwicklungsprozess hinweg einbezogen werden. Zu diesem Zweck ist es jedoch notwendig, die vielen Ideen und Meinungen der Community-Mitglieder zu bündeln. Da ein Entwickler-Team nicht die Gedanken von hunderten oder tausenden Personen mitverfolgen kann, sind somit Vermittlungsgremien einzuschalten, die schrittweise von Diskussion bis zur Entscheidungsfindung durch den Prozess führen.

Wie Füller et al. in ihrer Fallstudie feststellen, sind viele Kunden dazu bereit, eine virtuelle Plattform zu nutzen, um ihre Ideen in Entwicklungsprozesse einzubringen [FüBa+06]. Eine bereits etablierte Nutzer-Community mag beispielsweise mittels eines Forums verbunden sein. Dann könnte dieses auch zur Einbeziehung der Anwender genutzt werden. Ein besonderer Vorteil der Einbeziehung ganzer Nutzer-

Communities statt einzelner Anwender scheint die Möglichkeit zur Diskussion zu sein. Die Einbindung einer Gruppe einzelner Nutzer findet meist über Interviews und Fragebögen statt, die Auswertung erfolgt quantitativ. Im Gegensatz dazu besteht bei der Einbeziehung einer ganzen Nutzer-Community die Möglichkeit, dass die Anwender sich zuvor untereinander austauschen, gegenseitig über die Vor- und Nachteile von Alternativen informieren, diskutieren und somit eine bedachte Entscheidung treffen.

5.2 CDD – Konzept der strukturierten Community-Beteiligung

Damit eine Nutzer-Community in den Software-Designprozess einbezogen werden kann, benötigt diese zunächst einen eigenen Diskussionsbereich. Neben realen Treffen bieten sich dazu auch virtuelle Plattformen wie Foren an, in denen alle interessierten Anwender ihre Ideen und Meinungen austauschen und diskutieren können. Die Gruppe der dort beteiligten Nutzer bezeichnen wir als Volkskammer der Community. Bei hohem Interesse und somit eventuell zu ausufernder Diskussion kann die Anzahl der Volkskammer-Mitglieder auch durch die Entwickler begrenzt und ein Bewerbungsverfahren für die Interessierten eingeführt werden. Das zweite Gremium in diesem Konzept ist der sogenannte Zentralrat. Bestehend aus ausgewählten Anwendern sowie Mitgliedern des Entwickler-Teams ist dieser für die Verdichtung der Informationen und die endgültige Entscheidungsfindung zuständig. Da der Zentralrat somit eine sehr wichtige Rolle im Prozess übernimmt, sollten seine Mitglieder das Vertrauen sowohl der Anwender als auch der Entwickler genießen. Entsprechend dem Gatekeeper-Prinzip⁷ ist es somit sinnvoll, die im Zentralrat vertretenen Personen gemeinsam von Volkskammer und Entwickler-Team wählen zu lassen. Als private Nutzer haben die Zentralrats-Mitglieder die Möglichkeit, selbst an den Diskussionen der Volkskammer teilzunehmen. In ihrer übergeordneten Rolle verfolgen sie die Diskussion jedoch unparteiisch und greifen gegebenenfalls auch als Moderatoren ein. Die Nutzerschaft sollte im Zentralrat in der Mehrheit sein, um eine anwendergetriebene Entwicklung zu garantieren. In regelmäßigen Abständen veranstaltet der Zentralrat persönliche oder telefonische Konferenzen, um in zuvor festgelegten Themengebieten die Ideen und Interessen der Anwender gebündelt zu

⁷ Siehe Kapitel 3.4

diskutieren und gemeinsam Entscheidungen zu treffen. Die Ergebnisse der Zentralratssitzungen werden in einem für die Volkskammer-Mitglieder öffentlichen Bereich zusammengefasst. Somit entsteht ein zentrales Pflichtenheft, auf dessen Grundlage die Software-Entwickler arbeiten können. Möglichst früh sollte ein erster Prototyp erstellt werden, der kontinuierlich weiterentwickelt und allen beteiligten Anwendern zu Testzwecken zur Verfügung gestellt wird. Kernaspekte des Konzepts sind die durchgängige Einbeziehung der Anwender im kompletten Entwicklungsprozess, die Entwicklung am „lebenden Objekt“ sowie die endgültige Entscheidungskraft der Anwender, die durch deren Mehrheit im Zentralrat sichergestellt ist.

Das eingeführte Konzept des CDD lässt sich auch dahingehend erweitern, dass nicht nur eine, sondern gleich mehrere Nutzer-Communities einbezogen werden können. Jede dieser Gemeinschaften hat dazu einen eigenen Diskussionsbereich für ihre Volkskammer. Der Zentralrat besteht aus gewählten Vertretern jeder Volkskammer, die die Interessen ihrer eigenen Community vertreten. Der Zentralrat sollte auch in diesem Fall zum Großteil aus Anwendern gebildet werden.

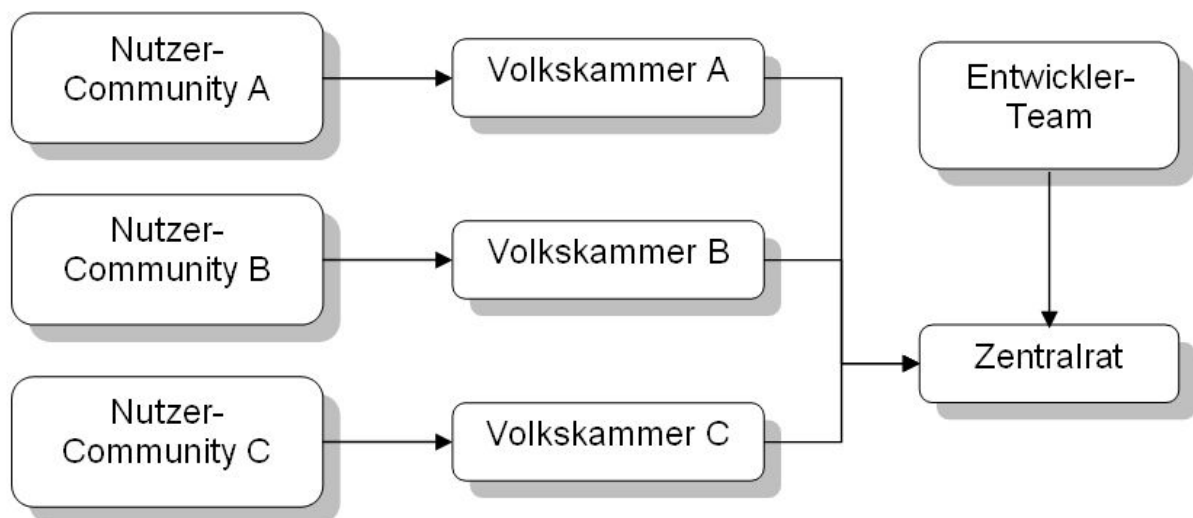


Abbildung 6: CDD-Konzept zur Einbeziehung mehrerer Nutzer-Communities

5.3 Forschungsfragen

Im Rahmen dieser Arbeit wurde die praktische Anwendbarkeit des vorgestellten Konzepts in einer Fallstudie genauer untersucht, um Potentiale, Vorteile und Risiken zu ermitteln. Es ergaben sich Forschungsfragen, die in drei Themengebiete geteilt

werden können: die Erforschung der Motivation aller Beteiligten an einem solchen Ansatz teilzunehmen, Maßnahmen zur technischen Unterstützung eines CDD-Prozesses sowie das eigentliche Konzept des CDD selbst.

Das vorgestellte Konzept des CDD basiert auf der Einbeziehung einer Nutzer-Community. Somit ist zu überprüfen, inwieweit der Community-Begriff unter den Anwendern selbst ausgeprägt ist und ob dies für sie einen entscheidenden Faktor bei der Zusammenarbeit darstellt. Weiterhin sollten die Gründe untersucht werden, die sowohl Anwender als auch Entwickler zur Mitwirkung bewegen. Die zweite Kategorie von Fragen betrifft die Möglichkeiten zur Unterstützung des CDD-Konzepts, um dieses auch für weitere Anwendungsfälle zu optimieren. In den Fragen zum allgemeinen Konzept soll geprüft werden, ob dieses Vorteile gegenüber anderen PD-Ansätzen bietet. Die zentrale Fragestellung dieser Arbeit betrifft somit die Ermittlung der Unterschiede zwischen der Einbeziehung einzelner Endanwender und einer etablierten Nutzer-Community.

Es ergaben sich somit die folgenden Forschungsfragen:

1. Motivation

- Wie ist das Selbstverständnis der Anwender für ihre Nutzer-Community?
- Was motiviert Nutzer, an einem CDD-Prozess zu partizipieren?
- Was motiviert Unternehmen, Nutzer stärker in diesen Prozess einzubeziehen?

2. Unterstützung des CDD-Ansatzes

- Wie kann man den Austausch zwischen der Community und den Designern geeignet unterstützen?
- Welche Tools sind geeignet, Nutzer kontinuierlich am Designprozess zu beteiligen?

3. CDD-Konzept

- Welche Vorteile bringt die Einbeziehung einer Nutzer-Community im Gegensatz zur Partizipation einzelner Endanwender?
- Wie wird das Ergebnis eines CDD-Konzepts von Nutzern und Entwicklern bewertet?

6 Fallstudie Omega

Zu den in Kapitel 5.3 gesammelten Forschungsfragen konnten sowohl die vorgestellten Fallstudien als auch eine weitreichende Literaturrecherche keine umfassenden Ergebnisse liefern. Das in Kapitel 5.2 neu entwickelte Konzept des CDD soll es jedoch ermöglichen, Antworten auf diese Fragen zu finden. Zu diesem Zweck und zur Evaluierung der Praxistauglichkeit des CDD-Ansatzes wurde das Konzept in einer Fallstudie empirisch untersucht.

6.1 Überblick über das Anwendungsfeld

Omega⁸ ist ein deutscher Softwarehersteller, der sich unter anderem mit der Entwicklung und Verbreitung von Produkten zur sinnvollen Vernetzung von PC und TV beschäftigt. Anhand des Omega-Projekts zur Entwicklung einer neuen Version des Media Centers *OmegaTV*, wurde exemplarisch untersucht, wie diese spezielle Nutzer-Community in den Designprozess einbezogen werden konnte.

6.1.1 Omega-Community

Der Anbieter Omega stellt seinen Nutzern einen eigenen Community-Bereich im Internet bereit. Dieser besteht aus einem Wiki-System, in dem Anwender ihr Wissen über die Omega-Produkte in strukturierter Form miteinander teilen, sowie einem Forum, in dem sich Nutzer untereinander sowie mit Omega-Mitarbeitern austauschen. Themen sind dabei einerseits Probleme und Erfahrungen mit den Produkten, andererseits jedoch auch Verbesserungsvorschläge für neue Produktversionen.

Über die letzten drei Jahre hat sich unter den über 15.000 registrierten Forumsnutzern (täglich gibt es durchschnittlich zehn Neuanmeldungen) eine aktive Community aus über 200 Mitgliedern gebildet, die sich regelmäßig an Forumdiskussionen beteiligen. Dabei machen die zehn aktivsten Nutzer rund ein Drittel der gesamten Kommunikation aus, unter ihnen befindet sich auch ein Mitglied des

⁸ Die Namen des Unternehmens und seiner Produkte wurden geändert.

Omega-Teams. Das Team besteht aus zehn Personen, die gemeinsam circa zwölf Prozent aller Forumsbeiträge liefern. Zu jedem dieser Postings wird die Zugehörigkeit zum Omega-Team als Zusatzinformation angezeigt, ebenso wie bei allen privaten Nutzern an gleicher Stelle deren entsprechende Benutzergruppe mitgeteilt wird. Dies kann entweder *Moderator/Moderatoren-Legende*, *Beta Tester* oder eine Kennzeichnung nach Anzahl der selbst geschriebenen Beiträge sein (*Silent Reader* bei keinen oder bis zu neun, *Newbie* bei zehn bis 24, *Member* bei 25 bis 99, *Expert* bei 100 bis 399, *Master* bei 400 bis 999 oder *Professional* bei über 1000 Beiträgen). Sowohl Moderatoren als auch Beta-Tester sind private Forumsnutzer, die sich freiwillig dazu bereit erklärt haben, zusätzliche Aufgaben zu übernehmen und dazu das Vertrauen des Omega-Teams genießen. Zu Projektbeginn gab es neun Moderatoren und elf Beta-Tester. Da bei den Nutzern eine so große Bereitschaft zur Beteiligung sowohl an Diskussionen als auch an freiwilligen Zusatzaufgaben besteht, bot sich gerade diese Community an, um zu erforschen, inwieweit deren Mitglieder auch in die Design- und Entwicklungsprozesse neuer Softwareartefakte und -versionen einbezogen werden können.

6.1.2 OmegaTV

OmegaTV ist eine Media Center Software für Microsoft Windows Systeme, deren erste Version im August 2004 erschien. Im Wiki-System von Omega werden die Funktionen vorgestellt: „*OmegaTV integriert alle Arten von Medien und Kommunikation wie z.B. Bilder, Musik, Videos, Internet, Telefonie und TV in einer zentralen Oberfläche*“. Den größten Vorteil gegenüber Konkurrenzsystemen stellt dabei die Flexibilität von OmegaTV sowohl bei der Verwaltung großer Bestände an Mediendaten als auch bei der Individualisierung der Anwenderschnittstelle dar.

Die Hauptfunktionen des neuen OmegaTV4 sollen die Wiedergabe von TV und Radio, eine elektronische Programmzeitschrift mit personalisiertem Videorekorder sowie die Verwaltung und Ausgabe von Videos, Musik und Fotos umfassen. Weitere Schwerpunkte sind eine hohe Flexibilität und die Möglichkeit zur Personalisierung des Systems. Alle weiteren Details und Systemfunktionen wurden mittels des CDD durch die Anwender mitbestimmt. Die Einbeziehung der zukünftigen Nutzer in den Entwicklungsprozess war somit ein entscheidender Ansatz für OmegaTV4. Bei der

Entwicklung der bisherigen Versionen 1 bis 3 wurden die Anwender nicht aktiv in den Designprozess involviert. Zwar wurden mehrere Vorschläge, die im Forum diskutiert wurden, gegebenenfalls in die neuen Systemversionen mit aufgenommen, jedoch lagen Planung und Entscheidung immer beim Omega-Team [vgl. Anhang - Interview 01].

6.1.3 Vorgehensweise zur Entwicklung von OmegaTV4

Bei der Entwicklung von OmegaTV4 kam das in Kapitel 5.2 vorgestellte CDD-Konzept zum Einsatz. Somit wurden die Anwender auf zwei Ebenen in die Steuerung des Entwicklungsprozesses eingebunden, um festzulegen, welche Funktionen wie und in welcher Priorität umgesetzt werden sollten. Durch das CDD sollten die alltäglichen Erfahrungen der Anwender *„direkt mit den Entwicklern und Produktverantwortlichen zu einem kundenorientierten Pflichtenheft verdichtet werden“* [Omega-Forum].

Die Volkskammer bestand in diesem Projekt aus 70 Nutzern, die im Omega-Forum ihre Ideen und Gedanken diskutierten. Moderiert wurde dies von den sieben Mitgliedern des Zentralrats, welche sich in vier ausgewählte Anwender und drei Mitglieder des Omega-Teams aufteilen ließen. In wöchentlichen Telefonkonferenzen stimmte der Zentralrat über die aktuellen Diskussionsthemen aus dem Forum ab. Die Ergebnisse dieser Gespräche wurden anschließend in das Wiki-System als zentrales Pflichtenheft übertragen und waren somit für jedermann sichtbar.

Auswahl der Mitwirkenden

Schon einige Monate vor Projektbeginn stellte das Omega-Team sowohl im Forum als auch im wöchentlichen Newsletter das Konzept der nutzereinbeziehenden Entwicklung der neuen OmegaTV Version vor. Die Mitglieder des Forums hatten vor dem Start der Entwicklungsarbeit die Möglichkeit, sich über ein Online-Formular für die Volkskammer oder den Zentralrat zu bewerben. Die ursprüngliche Planung hatte nur 30 Personen für die Volkskammer vorgesehen. Da jedoch alle Bewerber in ihren Anschreiben motiviert und verlässlich wirkten, wurde die Mitgliederzahl auf alle 70 Bewerber erhöht.

Die Mitglieder des Zentralrats hingegen wurden in Abstimmung des Omega-Teams und der Forumsmoderatoren ausgewählt. Die Moderatoren sind neun private Anwender, die sich freiwillig dazu bereit erklärt haben, im Forum für die Einhaltung verschiedener Grundsätze zu sorgen. Schon seit langer Zeit arbeitet Omega eng mit diesen Personen zusammen. Alle eingegangenen Bewerbungen für den Zentralrat wurden im internen Moderatoren-Forum vorgestellt. In einer freien Diskussion kamen Moderatoren und Omega-Team schnell zu einer Einigung. Die vier ausgewählten Anwender überzeugten durch Kompetenz, Diskussionsvermögen, sowie durch die Fähigkeit zur Vermittlung von Ideen und zur Deeskalation. Des Weiteren war es wichtig, dass die Zentralrat-Mitglieder ausreichend viel Zeit für die Arbeit am Projekt zur Verfügung stellen konnten. Diese Anforderungen führten schnell zu einer einstimmigen Entscheidung für die vier ausgewählten Anwender. Aus dem Omega-Team ergänzten der Produktmanager, der Projektleiter und der Qualitätsmanager den Zentralrat.

Projektverlauf

Der Verlauf des CDD lässt sich vom Start der Zusammenarbeit bis zur Veröffentlichung der Software in zwei Phasen unterteilen, die gemeinsam eine Dauer von neun Monaten umfassten. Vor dem offiziellen Projektbeginn gab es bereits eine etwa fünfmonatige Vorlaufzeit, in der Pläne für den Entwicklungsprozess erstellt wurden, und auch im Anschluss an die Veröffentlichung gab es eine weitere Zusammenarbeit der Entwickler und Anwender, so dass die Gesamtdauer der Software-Entwicklung mehr als 14 Monate betrug. Abbildung 7 stellt den Projektverlauf in Form eines Zeitstrahls dar.

Projektvorbereitungen: Mit den ersten Planungen für dieses Projekt begann Omega bereits im Januar 2007, fünf Monate vor dem offiziellen Projektstart. Im Rahmen dieser Vorbereitungen wurden grundlegende Vorgaben des Unternehmens für die neue Software festgelegt. Zu diesen gehörte der Direktzugriff auf alle Funktionalitäten über die One-Click-Navigation; die Startseite des neuen Systems sollte somit einen direkten Einstieg in die unterschiedlichen Medienwelten Fernsehen, Video, Bilder, Musik und Radio ermöglichen. Ebenfalls aufgrund des Architekturkonzeptes vordefiniert war das Konzept der Datenverwaltung über den

neuen Omega Media Service. Schriftlich erläutert wurden diese grundlegenden Anforderungen in der *Omega Bulle* [Anhang – Omega Bulle], die ebenfalls schon vor Projektbeginn veröffentlicht wurde.



Abbildung 7: Zeitstrahl des Projektverlaufs im Anwendungsfall Omega

Erste Projektphase: Start der aktiven Zusammenarbeit war der Kickoff-Workshop des Zentralrats am 1. und 2. Juni 2007. Das Ziel dieser Sitzung war es, die technischen Rahmenbedingungen und unabänderlichen Festlegungen der Entwickler vorzustellen

sowie ein grundlegendes Konzept für den Projektablauf zu entwickeln. Im darauffolgenden Wochenbericht wurde die im Workshop abgestimmte Vorgehensweise vorgestellt. Demnach war das Ziel des Projekts, „*die Entwicklungsarbeit am 'lebenden System' vorzunehmen, damit neben Volkskammer und Zentralrat auch alle anderen Anwender mitverfolgen (konnten), wie die Funktionen eingebaut (wurden)*“ [Omega-Forum]. Als Starttermin für die Aktivitäten der Volkskammer wurde die kurz darauf folgende Veröffentlichung der ersten *Technical Preview* vorgesehen. Diese sollte den Anwendern ein Bild von den schon festgelegten Prinzipien der Software vermitteln.

Die Diskussionen der Volkskammer fanden in einem separaten Unterforum statt, in dem nur die 70 Mitglieder Schreibrechte besaßen. Entsprechend den Vorgaben sollten alle Beiträge eine Kennzeichnung erhalten, die angab, ob es sich um eine Frage, Idee, Umfrage oder Diskussion handelte. Allerdings wurde dieses Vorgehen nicht konsequent durchgesetzt. Die einzelnen Threads enthielten meist zwischen fünf und 50 Beiträge, einer sogar über 300 Antworten. Bereits in den ersten drei Wochen wurden etwa 2000 Beiträge geschrieben, darunter sehr viele längere Postings [vgl. Anhang – Interview 06].

Auch die Mitglieder des Zentralrats hatten ihr eigenes Unterforum, das jedoch hauptsächlich für Terminabsprachen genutzt wurde. Ihre Ideen und Meinungen zur Produktentwicklung brachten auch diese Personen im Forum der Volkskammer ein. Jedes Zentralrats-Mitglied spezialisierte sich durch persönliche Interessen auf gewisse Themenbereiche. In diesen Gebieten verfolgte derjenige die Diskussionen stärker und fungierte im Volkskammer-Forum als Moderator. Wöchentlich erhielten alle Mitglieder des Zentralrats per E-Mail vom Produktmanager eine Zusammenfassung der aktuellen Forumdiskussionen, um eine umfassende Grundlage für die wöchentliche Telefonkonferenz zu haben. Die Konferenzen dauerten im Durchschnitt etwa zwei Stunden, in denen intensiv die zuvor festgelegten Themen diskutiert und Entscheidungen in den entsprechenden Bereichen getroffen wurden. Nach jeder Telefonkonferenz veröffentlichten die Mitglieder des Zentralrats ihre Ergebnisse im Wiki-System. Die dort gesammelten

Anforderungen bildeten somit das Pflichtenheft, nach dem die Entwickler anschließend das System implementierten.

Nach einer ersten Phase von etwa fünf Wochen waren die meisten Themen diskutiert und Beschlüsse gefasst. Somit setzten ab Mitte Juli 2007 die Diskussionen und gelegentlich auch die Telefonkonferenzen für eine Weile aus, während die Entwickler die erste Alpha-Version erzeugten.

Zweite Projektphase: Mit der Veröffentlichung dieser Alpha-Version begann Mitte Oktober, vier Monate nach dem Start der Zusammenarbeit, die nächste aktive Diskussionsphase. Innerhalb von sechs Wochen wurden insgesamt fünf Alpha-Versionen herausgegeben, die erste Beta-Version, die alle Module mit zumindest ihren Grundfunktionen enthielt, erschien am 3. Dezember 2007. Zum zweiten Workshop des Zentralrats im Januar 2008 waren auch die Moderatoren des Omega-Forums eingeladen, um den derzeitigen Entwicklungsstand zu besprechen. Ende Januar wurde ein Release Candidate der Software an die Mitglieder der Volkskammer herausgegeben. Bei Fertigstellung dieser Diplomarbeit war das Projekt noch nicht abgeschlossen, die Veröffentlichung der finalen Version war jedoch für den 25. Februar 2008, fast neun Monate nach Beginn der Zusammenarbeit, geplant.

Projektnachbereitungen: Mit der Veröffentlichung der Software sollte das CDD jedoch noch nicht abgeschlossen sein, sondern als fortlaufender Prozess betrachtet werden. Dem System sollten über Feature Packs weitere Funktionen hinzugefügt werden, deren Inhalte und Prioritäten ebenfalls in Absprache mit den Anwendern festgelegt werden sollten.

6.2 Untersuchungsdesign

Zur Untersuchung der im Rahmen dieser Arbeit begleiteten Fallstudie wurde die Technik *Action Research* angewendet. Dieser Begriff bezeichnet eine Forschungsmethodik, die bereits seit den 1940er Jahren in der Medizin und den Sozialwissenschaften eingesetzt wird [vgl. Bask99, S. 2]. Vor allem zum Ende der 1990er Jahre wurde diese Vorgehensweise auch bei der Entwicklung von Informationssystemen immer häufiger eingesetzt. Die Basis-Annahme des Action

Research ist, dass komplexe soziale Prozesse am Besten untersucht werden können, indem Veränderungen in diese Prozesse eingebracht und deren Folgen beobachtet werden [vgl. Bask99, S. 4]. Action Research existiert in verschiedenen Varianten, die sich je nach Anwendungsfall in ihrer Form unterscheiden. Viele Autoren sind sich darüber einig, dass diese unterschiedlichen Varianten stets vier Charakteristika erfüllen. Dies sind eine Orientierung auf Aktivitäten und Veränderungen, die Fokussierung auf einen Problemfall, ein systematischer, meist zyklischer Prozess, der Theorie und Praxis verbindet, sowie die Zusammenarbeit aller Teilnehmer [vgl. Bask99, PeRo84, HuLe80]. Im Idealfall ist auch der Forscher aktiv in die Vorgänge involviert, das erlangte Wissen ist sofort umsetzbar und es verspricht einen Vorteil sowohl für die Forschung als auch die betroffene Organisation. Meist basiert Action Research auf einem zyklischen fünfstufigen Modell bestehend aus den Phasen: Diagnose, Planung, Durchführung, Evaluation sowie Lernprozess [vgl. Bask99, S. 14-16].

6.3 Methodik und Interviewleitfäden

Entsprechend des in Kapitel 6.2 beschriebenen Untersuchungsdesigns bestand die Diagnosephase im untersuchten Anwendungsfall neben einer umfangreichen Studie der Fachliteratur aus der beobachtenden Begleitung des Projektverlaufs sowie Interviews mit unterschiedlichen Beteiligten. Sowohl Anwender als auch Entwickler wurden befragt, um eine Anforderungsanalyse für die Herangehensweise an ein nutzereinbeziehendes Entwicklungsprojekt durchzuführen. Im Mittelpunkt der Untersuchung stand die Erfassung der Motivation aller Beteiligten, ihrer Erwartungen und Vorbehalte an das Projekt, der praktischen Möglichkeiten zur Einbeziehung von Anwendern sowie der in der Praxis auftretenden Schwierigkeiten. Als Teil der Planungsphase wurde aus den ermittelten Erkenntnissen ein Maßnahmenkatalog zur Verbesserung der Zusammenarbeit von Entwicklern und Nutzern abgeleitet. In der Durchführungsphase wurde eine dieser Maßnahmen prototypisch umgesetzt. Die Auswahl der Maßnahme wurde in Absprache mit dem Omega-Team getroffen. Im Anschluss an eine kurze Testphase begann die Evaluierung des Prototyps. Diese fand wiederum durch Interviews mit Anwendern und Entwicklern statt. Nachdem die Untersuchung sich eingangs auf den konkreten Fall beschränkte, wurden nach einem Vergleich mit anderen Fallbeispielen Verbesserungsvorschläge für allgemeine

Anwendungsfälle abgeleitet. Diese Phase kann als der Lernprozess sowie gleichzeitig als Diagnosephase eines zweiten Zyklus betrachtet werden.

Interviews

Zu den befragten Interessengruppen gehörten nutzerseitig ausgewählte Anwender, Beta-Tester und Moderatoren des Forums. Im Unternehmen wurden Gespräche mit Entwicklern, Designern und einem Manager durchgeführt. Interviewt wurden acht Anwender und sechs Entwickler, alle Gespräche wurden zur späteren Analyse aufgezeichnet. Die Mitarbeiter des Omega-Teams und die Mitglieder des Zentralrats wurden in persönlichen Gesprächen befragt. Diese Interviews fanden entweder im Omega Büro oder bei den Workshops des Zentralrats statt. Die Mitglieder der Volkskammer hingegen wurden telefonisch befragt. Die Interviews dauerten jeweils zwischen sieben und 50 Minuten.

Mit jedem Interviewpartner fanden zwei Gespräche statt. In den Interviews zu Projektbeginn wurden die Omega-Community sowie die Voraussetzungen für das Projekt einführend untersucht. Die zweite Interviewreihe diente zum Einen der Evaluierung des im Rahmen dieser Arbeit erstellten Prototyps, zum Anderen der Untersuchung, inwiefern sich die Erwartungen der Teilnehmer erfüllt hatten und sich gegebenenfalls ihre Einstellung gegenüber dem CDD verändert hatte.⁹

6.4 Beobachtungen der ersten Projektphase

Vor allem zu Beginn fiel auf, dass die Anwender überaus motiviert und enthusiastisch in das Projekt starteten. Da alle Bewerber in ihren Anschreiben verlässlich wirkten, wurde die Mitgliederzahl der Volkskammer von geplanten 30 auf alle 70 Bewerber erhöht. Ein weiterer Grund, alle interessierten Personen in das Gremium aufzunehmen, war jedoch die Annahme der Mitarbeiter, dass nach einer ersten Begeisterungsphase die Motivation abebben könnte und sich somit einige Nutzer aus der Diskussion zurückziehen würden. Im Laufe des Projekts bestätigte sich diese Vermutung. In den ersten fünf Wochen herrschte eine sehr rege Beteiligung, bis alle entscheidenden Beschlüsse gefasst waren und ein erster Prototyp der Software

⁹ Die Fragenkataloge zu beiden Interviewphasen befinden sich im Anhang [Anhang – Interviews].

erwartet wurde. An dieser ersten Diskussionsreihe nahmen lediglich 49 der 70 Volkskammer-Mitglieder teil. Unter diesen war die Intensität der Beteiligung sehr unterschiedlich. Nach fünf Wochen hatte über die Hälfte der Teilnehmenden jeweils weniger als 15 Beiträge geschrieben, im Gegensatz dazu gab es jedoch zwölf Personen mit jeweils über 80 Postings, einer davon hatte sogar über 200 Beiträge geschrieben.

Als Motivation für ihre Teilnahme nannten einige Anwender den Gemeinschaftsgedanken und den Spaß an der Zusammenarbeit. Die meisten argumentierten jedoch mit der Möglichkeit, eigene Ideen in eine vielversprechende Entwicklungsarbeit einbringen zu können. Auch das Omega-Team erkannte, dass die Anwender durch ihre alltäglichen Erfahrungen neue Perspektiven und Anregungen liefern können. Vorherige Versionen der Software hatten zum Teil die Bedürfnisse des Marktes verfehlt, wodurch einige Features implementiert, jedoch hinterher kaum genutzt wurden. Dies sollte durch die Einbeziehung der Anwender in das Software-Design und somit eine direkte Ausrichtung an den Bedürfnissen der Nutzer verhindert werden. Eine gute Voraussetzung für das CDD war das beiderseitig gestärkte Vertrauensverhältnis durch positive Erfahrungen in der Zusammenarbeit des Teams mit Beta-Testern und Moderatoren. Besonders Teammanager und Pressesprecher sahen die Einbeziehung der Nutzer als „*eine absolut unerlässliche Sache*“ und „*zwangsläufige Entwicklung*“, die die Marktwirtschaft mit sich bringe [Anhang – Interview 02]. Andererseits waren sich die Mitarbeiter auch der Risiken eines solchen Verfahrens bewusst und begegneten dem Projekt somit mit unterschiedlichen Motivationsgraden. Während die Entwickler dem CDD eher skeptisch gegenübertraten, waren Team- und Produktmanager aufgeschlossener. Einer der Entwickler beschrieb ein „*flaues Gefühl, dass es ein bodenloses Fass werden könnte*“, wandte jedoch im gleichen Satz ein „*auf der anderen Seite (habe) man auch die Chance (...) gute Ideen und Anregungen zu kriegen*“ [Anhang – Interview 03]. Ein Risiko, das nahezu jeder der Mitarbeiter ansprach, war die zeitliche Problematik. Zwar erklärten die Verantwortlichen, es gäbe keine festen Fristen für den Projektablauf, betriebsintern waren jedoch Termine bekannt, die möglichst eingehalten werden sollten. Der so abzusehende Zeitraum von etwa sieben Monaten schien für eine Einigung und deren Umsetzung zu knapp bemessen. Eine große Zahl

teilnehmender Anwender erschwere dies noch mehr, die Entscheidungen, welche Systemfunktionen integriert werden sollten, würden im Endeffekt eventuell doch wieder bei den Entwicklern liegen, was zur Unzufriedenheit der Nutzer führen würde. Werde die Entscheidungsgewalt beim Anwender gelassen, so könnten Spannungen innerhalb der Gruppe große Probleme bereiten. Viele Meinungen und Standpunkte bieten ebenso viele Angriffsflächen, wodurch eine konzentrierte Arbeit erschwert werden könne. Um dies zu verhindern, sei möglicherweise eine Kontrollinstanz notwendig, die wiederum einen hohen Verwaltungsaufwand mit sich bringen könne. Ein Entwickler befürchtete, die Anwender würden sich nur auf neue Funktionalitäten, nicht auf eine sinnvolle Bedienbarkeit konzentrieren. Jedoch wurde bereits in den ersten Nutzer-Interviews deutlich, dass ihre Interessen diese beiden Gebiete ebenso abdeckten wie die Problematik der Systemstabilität. Zu hohe und naive Nutzererwartungen und daraus resultierender Frust, wenn Ideen der Anwender nicht berücksichtigt würden, wurden von der Projektleitung ebenso als Problemfeld angesehen wie die Voreingenommenheit mancher Entwickler. Auch wenn sie gewisse Risiken fürchteten, schätzten die Mitarbeiter das Projekt jedoch alle als eine Erfahrung ein, die zumindest lehrreich werden würde. Nicht abzusehen sei allerdings, wie hoch der Lerneffekt sei und ob es sich somit um eine lohnende oder doch eine Fehlinvestition handle. Im Idealfall könne das CDD ein perfektes System liefern, das am wahren Bedarf der Endanwender ausgerichtet sei und Omega somit eine klare Alleinstellung im Markt sichern könne [vgl. Anhang – Interview 05]. Aus Sicht der Anwender waren ohnehin kaum Risiken zu erkennen. Ein erfolgreiches Projekt brächte ihnen eine Software, die näher an den eigenen Wünschen und Anforderungen entwickelt wäre. Doch selbst ein Scheitern des CDD wäre aus Anwendersicht zwar als Verlust an Zeit anzurechnen, weitere Nachteile brächte es ihnen jedoch nicht.

Der von den Anwendern in den Interviews am häufigsten kritisierte Punkt war die mangelnde Kommunikationsbereitschaft des Omega-Teams, die die Nutzer mit fehlender Motivation der Mitarbeiter begründeten. Praktisch war jedoch der Mangel an sowohl Personalressourcen als auch Zeit ausschlaggebend für dieses Phänomen, das in der ersten Projektphase unterschiedliche Schwierigkeiten zur Folge hatte. Ein wesentliches Problem des Zentralrats bestand in der Verdichtung der vielen Ideen

und Meinungen der Volkskammer zu einer eindeutigen Liste von Entscheidungen. Vor allem zu Beginn war es schwierig, auch nur annähernd mit der Geschwindigkeit der Forumdiskussion mitzuhalten, da dem Zentralrat wöchentlich nur etwa zwei Stunden zur Entschlussfassung zur Verfügung standen. In den ersten Telefonkonferenzen wurde dieses Problem zusätzlich dadurch verstärkt, dass keines der Zentralrats-Mitglieder die Verantwortung für Entscheidungen übernehmen wollte; somit wurde zwar viel diskutiert, aber es wurden nur wenige klar formulierte Beschlüsse getroffen. Nach den ersten Wochen trat dieses Problem nicht mehr auf, jedoch wurde es erneut schwierig, als beim krankheitsbedingten Ausfall des Projektmanagers kein anderes Zentralrats-Mitglied bereit war, stellvertretend die Einträge im Wiki-System vorzunehmen. Durch die mangelnde Bereitschaft diese Veröffentlichungen zu verfassen, dauerte es oft sehr lang, bis alle Ergebnisse einer Telefonkonferenz eingetragen waren. In einigen Fällen waren sich die Zentralrats-Mitglieder im Nachhinein uneinig, ob die Einträge auch wirklich den Beschlüssen entsprachen, da diese schon zu lange zurücklagen. Ein weiterer negativer Effekt der langen Wartezeiten war, dass die Mitglieder der Volkskammer länger uninformiert blieben und sich deshalb ausgeschlossen fühlen. Zum Ende der ersten Projektphase schien nahezu die gesamte Volkskammer diese Befürchtung zu teilen. Sowohl in Interviews als auch im Unterforum erläuterten die Anwender, sie fühlten sich aus dem eigentlichen Designprozess ausgeschlossen, Zentralrat und Omega-Team würden die Meinungen und Diskussionen der Volkskammer bei den Entscheidungen gar nicht weiter beachten, die Volkskammer hätte somit nur eine Alibi-Funktion [vgl. Anhang – Interview 11 und 12]. Dieser Eindruck verstärkte sich, als auch die Wartezeit auf die erste Alpha-Version immer wieder verlängert wurde und die Volkskammer-Mitglieder keine neuen Informationen erhielten. *„Ich hätte mir viel schneller viel mehr frisches Material gewünscht. (...) Es hätte eigentlich viel schneller etwas umgesetzt werden müssen, um einfach den Kessel am Dampfen zu halten. Immer wieder Futter nachzuliefern, damit die Leute was zu reden haben“* [Anhang – Interview 12]. Ratlosigkeit und Unzufriedenheit der Anwender waren die Folge. Jedoch lag die Schuld an der Uniformiertheit der Anwender nicht allein bei Omega-Team und Zentralrat, viele Nutzer schienen nur selten oder überhaupt nicht die Beiträge im Wiki-System zu lesen. Es gab während der gesamten Wartezeit auf die erste Alpha-Version keinerlei Beschwerden über die dort festgelegten Beschlüsse.

Als jedoch die Alpha-Version veröffentlicht wurde, wurden viele Punkte kritisiert und bemängelt, die schon seit Langem im zentralen Dokument einzusehen waren.

6.5 CDD-Unterstützung

Aufgrund der Analyse der ersten Interviewreihe ließen sich unterschiedliche Schwierigkeiten bei der Kommunikation zwischen Entwicklern und Anwendern erkennen. Vor allem die Mitglieder der Volkskammer beschwerten sich über mangelnde Auskünfte, selbst zum aktuellen Entwicklungsstand erhielten sie kaum Informationen. Ein weiteres Problem war die schwache Verbindung zwischen Produkt und Diskussion. Würde diese verstärkt, so könnte dies gegebenenfalls auch die Übersichtlichkeit erhöhen und somit die Anwender eher zufriedenstellen. Die folgenden Ansätze verfolgen diese Ziele.

Ansatz 1: Einbindung der Diskussion in das Produkt

Idee: Fragen zu Funktionen und Bedienung treten ebenso wie Verbesserungsideen meist während der Systemnutzung auf. Eine unkomplizierte und schnelle Möglichkeit, seine Gedanken an die Entwickler weiterzuleiten oder sogar allgemein zu veröffentlichen, könnte für viele Anwender ein weiterer Anreiz sein, sich an Diskussionen zu beteiligen.

Im Fall von OmegaTV ist es beispielsweise denkbar, ein Browser-Modul in die Software einzubinden, über das der Anwender direkt ins Omega-Forum gelangen kann. Eine weitere Idee wäre ein Feature, das dem Nutzer aus jedem Systembereich heraus ermöglicht, kurze Nachrichten an das Entwickler-Team zu senden. Diese Meldungen sollte dabei implizit die Information enthalten, aus welchem Modul sie gesendet wurden, gegebenenfalls sollte auch ein Screenshot angehängt werden können.

Mögliche Schwierigkeiten: Ein solcher Ansatz ist im Bereich von Media Center Software schwierig. Bereits in der vorherigen Version OmegaTV2 wurde ein Browser-Modul integriert, welches dem Anwender den Zugriff auf das Omega-Forum ermöglichte. Dieses wurde jedoch kaum genutzt. Die Anwender von OmegaTV neigen meist zu einem ausschließlich der Entspannung dienenden Gebrauch der

Software. Systemkonfigurationen werden vorab am PC vorgenommen, so dass die hauptsächliche Nutzung des Systems eher passiv erfolgt. Des Weiteren dient in den meisten Fällen nur eine einfache Fernbedienung der Eingabe, wodurch die Möglichkeiten ebenfalls eingeschränkt werden.

Meinungen der Beteiligten: In der zweiten Interviewreihe wurden sowohl Entwickler als auch Anwender zu den hier vorgestellten Ideen befragt. In Bezug auf eine in die Software eingebaute Feedback-Möglichkeit waren die Meinungen sehr wechselhaft. Die Entwickler schätzten die Maßnahme zwar als „*sinnvoll*“, „*hilfreich*“ und „*interessant*“ ein, merkten jedoch an, dass dieser Ansatz einen hohen Betreuungsaufwand mit sich bringe, den Omega aus Mangel an personellen Ressourcen nicht erfüllen könne. Auch die Anwender bewerteten die Idee als grundsätzlich positiv, hatten jedoch große Zweifel, ob eine solche Möglichkeit von vielen Anwendern genutzt werde. Die Steuerung über eine Fernbedienung sei dabei ebenso ein Problem wie die Angst vieler Nutzer, durch das automatische Versenden von Information persönliche Daten preiszugeben.

Ansatz 2: Einbindung des Produkts in die Diskussion

Idee: Eine weitere Möglichkeit, Software und Diskussion stärker zu verbinden, ist die graphische Anpassung der Kommunikationsplattformen an das Produkt.

Im Fall von OmegaTV könnte beispielsweise das Unterforum der Volkskammer mit einem Design ähnlich dem OmegaTV Startbild beginnen. Wie für die Funktionalitäten der Software könnten darüber die einzelnen Diskussionsbereiche erreicht werden, die sich mit diesen Funktionen befassen. Ein solcher Bereich würde dann ähnlich der bisherigen Ansicht eine Übersicht der aktuellen Topics enthalten, jedoch wären diese auf den jeweiligen Funktionsbereich beschränkt. Weitere Zusatzfunktionen könnten die Übersichtlichkeit ebenfalls erhöhen. So wäre es möglich, dass jede Antwort innerhalb eines Threads mit einer Pro- oder Contra-Bewertung versehen werden muss. Dies könnte Auswertungen vereinfachen und auch in der Themenübersicht eine gute Visualisierung der gesammelten Meinungen widerspiegeln.

Ein ähnlicher Ansatz wird bei der Städteplanung in einigen geographischen Communities verfolgt [vgl. PiMä+00]. Ein Community-Design-Tool hat dabei die Aufgabe, sowohl die Repräsentation des Design-Modells als auch den Austausch zwischen Bürgern und Designern zu fördern. Dem Anwender werden mehrere zwei- und dreidimensionale Sichtweisen auf die Pläne ermöglicht, um auch komplexere Aspekte zu verstehen. In einer Diskussion können Threads mit Referenzen zu bestimmten Modellaspekten versehen werden, einzelne Beiträge mit einer Stellungnahme für oder gegen diesen Punkt. Abbildung 8 zeigt ein solches Beispiel, das Forum des Community-Design-Tools Zeno.

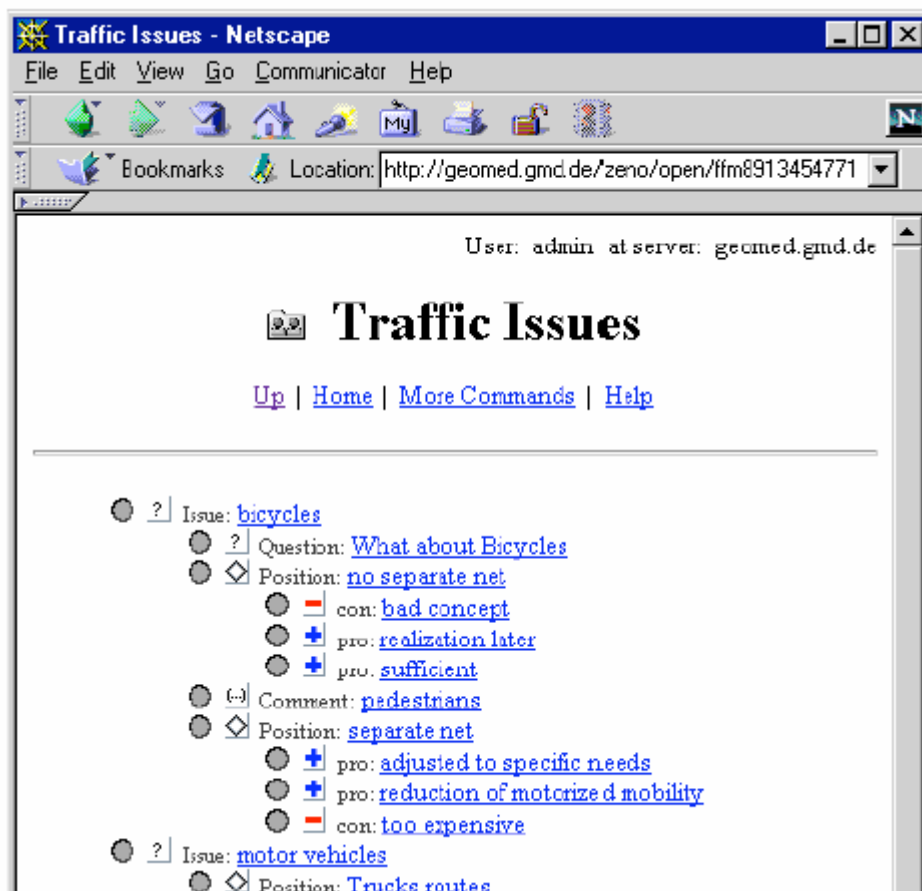


Abbildung 8: Beispielforum des Community-Design-Tools Zeno [PiMä+00]

Auch im Wiki-System könnte eine graphische Anpassung vorgenommen werden. Ähnlich den zuvor beschriebenen Veränderungen wäre es ebenso dort möglich, die Strukturen der Software abzubilden und somit einen Wiedererkennungswert zu schaffen. Vor allem für Neukunden wäre die Abbildung der Bedienstruktur hilfreich, um ihnen die Einarbeitung in die Software zu erleichtern. Eine gute Suchfunktion

sollte jedoch zusätzlich integriert sein, damit Anwender auch Informationen zu Systemfunktionen erhalten, von denen sie nicht wissen, wie diese zu erreichen sind.

Mögliche Schwierigkeiten: Probleme könnten insofern auftreten, dass die Nutzer eventuell keine Veränderungen ihrer gewohnten Diskussionsumgebung wünschen. Eine ausschließlich im Volkskammer-Bereich erneuerte Visualisierung kann das Gesamtbild des Forums unterbrechen und somit entweder störend wirken oder bei den Anwendern den Wunsch nach einer Anpassung des gesamten Forums hervorrufen. Auch bei einer partiellen Veränderung des Wiki-Systems könnte dieser Wunsch nach einer generellen Erneuerung auftreten.

Meinungen der Anwender: Fast alle Befragten sprachen sich für eine Veränderung der Kommunikationsplattformen aus, konkrete Vorstellungen hatte dabei jedoch kaum jemand. Eine graphische Anpassung des Forums an die Software bewerteten die Entwickler sehr unterschiedlich, die Meinungen reichten von „charmant“ und „modern“ bis hin zu der Aussage: „dann würde es wohl eher nerven, als dass es Sinn macht“. Problematisch in Bezug auf den Fall Omega empfanden sie dabei vor allem die Komplexität und Anpassbarkeit der Software, die sich somit ebenso im Forum widerspiegeln würde. Bei einer einfachen Software-Struktur hingegen sei der Ansatz sinnvoll. Die Anwender schätzten die Idee zwar als vorteilhaft ein, vermuteten jedoch, dass die Umsetzung schwierig sein könne. Da sie bei Verfolgung dieses Ansatzes eine einheitliche Anpassung des ganzen Forums wünschten, sei der Aufwand für eine solche „Spielerei“ zu groß. Ein Umbau des Wiki-Systems, mit dessen Struktur die Nutzer bereits vor Projektbeginn unzufrieden waren, wurde sowohl von Anwendern als auch Entwicklern stark befürwortet. Es sei ein „erfolgversprechender Weg“, allerdings müsse dafür gesorgt werden, dass das System mit mehr Informationen gefüllt werde, als es bisher der Fall sei.

Ansatz 3: Veränderung der Kommunikationsplattformen

Ansatz 3.1: Bündelung von Diskussion und Information

Idee: Im vorliegenden Fall kamen zur Unterstützung der Kommunikation die beiden Plattformen Forum und Wiki-System zum Einsatz. Jedoch wurde im Projektverlauf ebenso wie in den Interviews schnell deutlich, dass die angewandte Unterteilung von

Diskussion und Information auf diese beiden Plattformen nicht die beste Lösung war. Vor allem von Seiten der Anwender wurde immer wieder der Wunsch geäußert, beide Bereiche in einem System zu bündeln. Ein gewöhnliches Wiki-System eigne sich dazu besser als ein Forum, da es ohnehin neben dem Informationsteil über einen abgetrennten Diskussionsbereich verfüge.

Mögliche Schwierigkeiten: Würde man das Wiki-System gleichzeitig zu Information und Diskussion nutzen, so könnte es schwierig werden, die Diskussionen übersichtlich zu halten. Ein Wiki-System verfügt nicht über alle administrativen Strukturen eines Forums, auch das Verfassen von Beiträgen wird dort oft als zu kompliziert betrachtet. Neue Beiträge sind nicht leicht aufzufinden. Es müssten somit zunächst weitere Veränderungen an der Wiki-Struktur vorgenommen werden, um einen guten Diskussionsfluss zu ermöglichen.

Ansatz 3.2: Community-Gruppendiskussion

Idee: Ein speziell in der Omega-Community auftretendes Problem ist, dass die Bereitschaft zur Beteiligung im Forum vor Beginn des Projekts stark abgenommen hatte. In den Interviews erklärten viele Anwender, dass in dieser Community ein wahres Gemeinschaftsgefühl nur unter wenigen Mitgliedern herrsche, die seit vielen Jahren im Forum aktiv sind. Eine Ursache für die schwindende Beteiligung ist laut einigen Nutzern die mangelnde Rückmeldung durch das Omega-Team.

Ein Ansatz, der dazu dienen kann, Vertrauen zu gewinnen und somit eine Neubelebung des Forums anzustreben, ist eine Gruppendiskussion. Im Fall von OmegaTV könnte die Volkskammer ihre Ideen in einer gemeinsamen Diskussion austauschen, welche von einigen Mitgliedern des Zentralrats moderiert würde. Durch ein direktes Feedback fühlen sich die Anwender vermutlich eher ernst genommen.

Mögliche Schwierigkeiten: Das größte Problem bei diesem Ansatz ist die Notwendigkeit der freiwilligen Beteiligung und zeitlichen Abstimmung. Es müssen sich viele Anwender finden, die zu einem festgelegten Zeitpunkt miteinander diskutieren. Bei sehr hoher Beteiligung müssen die Moderatoren eventuell insofern für eine bessere Übersichtlichkeit sorgen, indem sie Flüsternachrichten der

Teilnehmer sammeln und geordnet in die Diskussion einstreuen. Auf diese Weise können die Diskussion sowie die anschließende Auswertung einen hohen organisatorischen und zeitlichen Aufwand hervorrufen.

Meinungen der Beteiligten: Gruppendiskussionen hielten alle befragten Personen für ein sehr sinnvolles Mittel. Noch beliebter seien reale Treffen. Diese würden zwar einen erheblich größeren Organisationsaufwand verursachen, allerdings lohne sich dieser, da bei einem direkten Zusammentreffen viel effektiver gearbeitet werden könne. Auch bei solchen Workshops sei eine Moderation durch die Zentralrats-Mitglieder sinnvoll. Weiterhin sei zu beachten, dass die Teilnehmer nicht aufgrund ihrer regionalen Nähe, sondern aufgrund ihrer Interessen bestimmten modul-orientierten Workshops zugeteilt würden [vgl. Anhang – Interview 19]. Auch die Omega-Mitarbeiter befürworteten diese Idee von Brainstorming-Veranstaltungen, in denen alle Ideen mit ihren eventuellen Randbedingungen detailliert ausgearbeitet werden sollten. Solche Treffen seien sehr vielversprechend, der zeitliche und organisatorische Aufwand jedoch sehr groß, so dass die Planung dieser Veranstaltungen frühzeitig beginnen sollte. Sei die Organisation solcher Treffen nicht möglich, so wären Webkonferenzen eine weitere Alternative.

Ansatz 3.3: Einsatz eines kontextabhängigen Chat-Tools

Idee: Ein Ansatz für längerfristige Projekte ist der Einsatz kontextabhängiger Chat-Tools wie beispielsweise *ConcertChat*¹⁰. Dieses Tool ermöglicht mehreren Personen eine gemeinsame Diskussion, wobei neben dem textuellen Austausch auch Dokumente und Grafiken miteinander geteilt werden können. Zur besseren Verdeutlichung von Zusammenhängen können Beiträge mit einer Markierung versehen werden, die angibt, auf welchen vorherigen Beitrag, welchen Teil einer Skizze oder eines gemeinsamen Dokuments sie sich beziehen. Ein weiterer Vorteil von ConcertChat ist die persistente Speicherung der Beiträge, wodurch auch eine asynchrone Diskussion ermöglicht wird.

¹⁰ http://www.ipsi.fraunhofer.de/concert/index_de.shtml?projects/chat

Bei Omega könnten so beispielsweise auch die Mitglieder der Volkskammer in regelmäßigen Sitzungen Design-Entwürfe diskutieren. Die Größe der Gruppe sollte dabei jedoch überschaubar bleiben, um einerseits Terminabsprachen zu erleichtern, andererseits ein zu großes Durcheinander bei der Diskussion zu vermeiden. Somit sollte zu speziellen Chat-Terminen jeweils ein zusammenhängendes Themengebiet diskutiert und der Termin mit der Gruppe der sich dafür verstärkt interessierenden Anwender besprochen werden.

Mögliche Schwierigkeiten: Problematisch werden kann die Terminabsprache jedoch auch in diesem Fall. Bei den wöchentlichen Telefonkonferenzen des Zentralrats (also einer Gruppe aus lediglich sieben Personen) waren oft mehrere Teilnehmer verhindert. Die Absprache des Termins für ein gemeinsames Treffen des Zentralrats brachte erst nach über zwei Monaten ein Ergebnis, doch selbst dieser Termin wurde kurzfristig wieder abgesagt.

6.6 Beobachtungen der zweiten Projektphase

Drei Monate nach Ende der ersten Diskussionsphase wurde die von den Anwendern lang ersehnte erste Alpha-Version der Software veröffentlicht und somit allen interessierten Nutzern zu Tests zur Verfügung gestellt. Entgegen der ursprünglichen Planung enthielt diese Alpha-Version jedoch nur einen stark eingeschränkten Funktionsumfang, da der Aufbau der grundlegenden Technologie viel Zeit gekostet hatte. Somit waren zunächst lediglich die Module *TV*, *Radio*, *Video* und *EPG* mit ihren Grundfunktionen integriert.

Bei den Anwendern führte die späte Veröffentlichung dieser Alpha-Version zu sehr unterschiedlichen Reaktionen. Im öffentlichen Forumsbereich überwiegte ein positiver Eindruck, der den Anwendern Vorfreude auf die Endversion bereitete. Innerhalb der Volkskammer hingegen war die Situation sehr angespannt, mancher behauptete sogar, das CDD habe versagt. Die geringe Zahl an offensichtlichen Veränderungen gegenüber der Technical Preview enttäuschte viele Volkskammer-Mitglieder ebenso wie der optische Eindruck der Alpha-Version, woraufhin einige Nutzer ihr bereits zuvor erwachtes Misstrauen bestätigt sahen und sowohl dem Zentralrat als auch dem Entwickler-Team unterstellten, die Wünsche der Nutzer

komplett ignoriert zu haben. Die Diskussion drohte zu eskalieren, manche Zentralrats-Mitglieder sahen sich persönlich angegriffen und so gingen auch sie in die Offensive. Es kam zu Schuldzuweisungen zwischen Volkskammer und Zentralrat. Offensichtlich war, dass die Kommunikation zwischen den beiden Gremien an einigen Stellen gescheitert war. Nachdem sich beide Parteien ein wenig beruhigt hatten, konnte die Diskussion wieder friedlich fortgeführt werden. Dies lag vor allem an den schnellen Reaktionen des Omega-Teams auf die Beschwerden der Nutzer, sowohl in Form von Stellungnahmen über das Forum als auch durch Anpassungen an der Software.

Doch auch weiterhin konnte der Verdacht mancher Nutzer, Omega wolle nur seine eigenen Vorstellungen durchsetzen, nicht vollständig ausgeräumt werden. *„Die haben mal versucht, das Ding so ein bisschen jetzt beeinflussen zu lassen, aber trotzdem immer wieder noch das eigene Bild vor den Augen gehabt“* [Anhang – Interview 19]. Dies lag überwiegend an zwei Faktoren. Bereits zu Beginn des Projekts waren seitens des Omega-Teams einige Aspekte festgelegt, die vor allem die Möglichkeiten im Bereich Design und Bedienstruktur stark einschränkten. Aus Sicht des Unternehmens war dies sinnvoll, um den Anwendern einen ersten optischen Eindruck zu vermitteln, der ihre Kreativität weiter anregen sollte. Viele Anwender legen jedoch gerade auf diese Gebiete großen Wert und äußerten viele Vorschläge und Ideen, die in der ersten Projektphase weitestgehend unbeachtet blieben. So erläuterte auch ein Omega-Mitarbeiter: *„Da wurde teilweise gar nicht richtig drauf eingegangen und das hat natürlich für eine Zeit lang die Fronten auch ein bisschen verhärte“* [Anhang – Interview 14]. Zu Beginn der zweiten Projektphase wurden diese Diskussionen neu aufgegriffen und es wurde eine anpassbare Benutzeroberfläche geschaffen. Entgegen der Vermutungen der Anwender war diese bereits von Beginn an eingeplant, was auch im Wiki-System vermerkt, jedoch ansonsten nicht kommuniziert worden war.

Der zweite Punkt, der die Anwender an ihrem Einfluss zweifeln ließ, war die Tatsache, dass ein Großteil der umgesetzten Vorschläge Ideen des Produktmanagers waren. Dieser hatte seine Konzepte jedoch in die Volkskammer-Diskussion eingebracht wie jeder Anwender es auch konnte. Somit waren es

Mehrheitsbeschlüsse der Nutzer, die letztendlich umgesetzt wurden. Er selbst bestätigte im Interview, die Entwicklung wäre für ihn leicht steuerbar gewesen, da er statt undurchdachter Ideen schlüssige Konzepte vorstellte, zu denen sich kaum kontroverse Diskussionen entwickelten. Dies entspreche jedoch nicht dem Prinzip des CDD, weshalb es eventuell nötig sei, sich als Mitglied des Entwickler-Teams mehr aus der Ideenfindung herauszuhalten [vgl. Anhang – Interview 17]. Dann allerdings, so erklärten sowohl sein Team-Kollege als auch ein Anwender, wären zu wenig durchdachte Ideen in die Entwicklung eingeflossen. Somit habe sich die Einbringung des Produktmanagers aus der Notwendigkeit entwickelt, da die übrigen Volkskammer-Mitglieder zu zurückhaltend gewesen seien.

Auch innerhalb des Zentralrats kam es zu Schuldzuweisungen zwischen Anwendern und Omega-Mitarbeitern. So beschwerten sich die Mitarbeiter darüber, dass die meisten Aufgaben des Zentralrats, darunter vor allem die zeitaufwendige Pflege des Wiki-Systems, ihnen überlassen worden seien und sie somit noch weniger Zeit gehabt hätten, sich mit ihren eigentlichen Aufgabengebieten zu befassen. Alle Zentralrat-Mitglieder waren sich in den Interviews darüber einig, dass eine klare Aufgabenverteilung innerhalb ihres Gremiums gefehlt hatte. So fühlten sich die Omega-Mitarbeiter zum Teil im Stich gelassen, während die Anwender das Gefühl hatten, selbst innerhalb des Zentralrats vom eigentlichen Entscheidungsprozess ausgeschlossen zu sein. *„Manchmal kam man sich überhaupt nicht informiert vor und wir waren eigentlich eine interne Gruppe (...) und wenn es einen (...) Meilenstein gibt, der gefordert wird von der finanziellen Etage, sag ich mal, dann kann das uns auch gesagt werden und nicht drumrum geredet werden“* [Anhang – Interview 21]. Das grundsätzliche Problem lag somit auch in diesem Zusammenhang an fehlender Kommunikation, jedoch mangelte es auch an einem grundlegenden Konzept und einer Moderation seitens der Projektleitung. Selbst die Omega-Mitarbeiter im Zentralrat beschwerten sich darüber, *„dass es (...) keine Richtlinien gab, dass auch ein Zentralkammer-Mitglied (...) gar nicht wusste, was es zu tun hat“* [vgl. Anhang – Interview 14].

Ein weiterer Problempunkt, der ebenfalls das Kommunikationsverhalten aller Beteiligten betraf, war die Nutzung des Wiki-Systems. In diesem sollten alle

Beschlüsse des Zentralrats als zentrales Pflichtenheft zusammengefasst werden, damit jeder Nutzer jederzeit den aktuellen Stand der Entschlussfassung verfolgen konnte. Jedoch sahen nur sehr wenige Anwender regelmäßig in dieses Dokument. So beschrieb der Produktmanager: *„Es hat sich niemand die Mühe gemacht, sich dahin zu setzen und das wirklich zu lesen. Ich hab zwischendurch immer wieder nachgefragt und es kam auch immer wieder das Signal `Ja - ist wirklich in Ordnung` und dann, als die ersten Versionen draußen waren, kamen dann die Unkenrufe“* [Anhang – Interview 17]. Obwohl es bis Ende November fast 120.000 Zugriffe auf das Wiki-System gab, wurde darauf kaum reagiert. Wurden jedoch die gleichen Inhalte im Forum veröffentlicht, entwickelten sich große Diskussionen. Der Medienbruch zwischen Forum und Wiki-System schien ein Problem zu sein. Auch die Anwender begründeten ihr Verhalten damit, dass es keine Benachrichtigungen gab, wann und zu welchen Themen neue Beiträge geschrieben wurden. Zwar verfügt das Wiki-System über eine Änderungshistorie, das Pflichtenheft war jedoch in einem zusammenhängenden Beitrag verfasst, so dass die Anwender diesen komplett durchsuchen mussten, um Neuerungen zu finden. Somit konnte die vorhandene Unterstützungsmaßnahme an dieser Stelle nicht genutzt werden. Da das Verfassen der Wiki-Einträge durch die mangelnde Aufgabenverteilung innerhalb des Zentralrats sehr schleppend voran ging, mieden viele Nutzer dieses Medium und konzentrierten sich allein auf die Forumdiskussionen. Dort jedoch hielt der Zentralrat es nicht für nötig, seine Entschlüsse zu veröffentlichen, da zu diesem Zweck das Wiki-System dienen sollte. Die Tatsache, dass nur sehr wenige Anwender die Entwicklung des Pflichtenheftes regelmäßig verfolgten, wurde erst mit Erscheinen der ersten Alpha-Version offensichtlich. Viele Nutzer beschwerten sich über Aspekte der Software, die schon seit Monaten in genau dieser Form im Wiki-System beschrieben waren. Die Mitglieder des Zentralrats reagierten auf diese Beschwerden mit Wut und Enttäuschung. Einige Volkskammer-Mitglieder hingegen sprachen weiterhin davon, die Zentralratsbeschlüsse spiegelten nicht die Meinung der mitwirkenden Nutzer wider, was sie vor allem erneut auf die Bereiche Design und Bedienkonzept bezogen. Ein Volkskammer-Mitglied fasste die Situation folgendermaßen zusammen: *„Ich verstehe die ganze Aufregung nicht. 1. Es steht alles im Wiki und es sieht genauso aus wie im Wiki. 2. Ich habe nirgends Beschwerden gelesen über das, was im Wiki steht. (...) 8. Dann lasst uns das Wiki ändern“* [Omega-Forum]. Ähnlich

reagierte auch der Zentralrat, der nach der nächsten Telefonkonferenz die Nutzer schnell über den aktuellen Entwicklungsstand informierte. Auch Negativ-Entscheidungen wurden in diesem Bericht ausführlich begründet. Des Weiteren wurde darum gebeten, die Diskussion in strittigen Punkten wieder aufzunehmen und die Einträge im Wiki-System zu überprüfen.

In der zweiten Projektphase verlagerte sich ein Großteil der Diskussionen in den öffentlichen Forumsbereich. Somit konnten alle Anwender ihre Erfahrungen mit den Alpha- und Beta-Versionen der Software einbringen. Neben den Volkskammer-Mitgliedern beteiligten sich an der offenen Diskussion 214 weitere Anwender, jedoch beschränkte sich der Großteil darauf, Programmfehler zu melden. Lediglich etwa 40 Personen beteiligten sich regelmäßig, schrieben Beiträge zu allgemeinen Aspekten, wie beispielsweise der Bedienstruktur, und nannten weitere Verbesserungsideen. Obwohl jede Alpha-Version etwa 500 Mal heruntergeladen wurde, nahm die Zahl der Reaktionen von der ersten bis zur fünften Version stark ab. Nach den zum Teil sehr emotionsgeladenen Beiträgen zur ersten Alpha-Version, führt die schnelle Umsetzung neuer Beschlüsse zu mehr Zufriedenheit bei den Anwendern. Bei den beiden letzten Alpha-Versionen beschränkten sich die Reaktionen jedoch fast ausschließlich auf gelegentliche Fehlermeldungen. Dies führte wiederum zur Unzufriedenheit des Omega-Teams, da die Entwickler nicht wussten, ob die Anwender mit der Software zufrieden waren.

Ein ebenfalls komplizierter Aspekt des CDD ist die längerfristige Zusammenarbeit vieler Personen, die sich und eine solche Arbeitsweise kaum kennen. Oft wurden die Diskussionen sehr emotionsgeladen. Ein Volkskammer-Mitglied beschrieb dies: *„Eine Community lebt ja von der Diskussion, sonst wär es ja keine. Da kann's ja schon mal hitzig zu gehen, letztendlich werden sich doch hoffentlich auch wieder alle vertragen“* [Omega-Forum]. Durch die zunächst eher negativen Reaktionen der Volkskammer auf die erste Alpha-Version kam es seitens eines Omega-Mitarbeiters zeitweise sogar zu Aussagen wie, *„damit hat sich eigentlich das Thema (CDD) für mich erledigt. Ich schließe damit jetzt gedanklich ab, es war der größte Fehler den Usern die Hand auszustrecken“* [Omega-Forum]. Jedoch beruhigte sich auch dieser schnell wieder. Im Interview gab er an, er sei zwar mit dem Verlauf des Projekts nicht zufrieden, das CDD sei jedoch grundlegend eine vielversprechende Technik, die er

gerne erneut anwenden würde [vgl. Anhang – Interview 14]. Während die Anwender aus der Volkskammer anfangs sehr gereizt und emotional auf die Alpha-Versionen reagierten, herrschte im öffentlichen Bereich des Forums viel mehr Begeisterung. Viele Aspekte der Software wurden gelobt. Negativ angemerkt wurden zunächst nur Systemfehler, auf die jedoch auch schnell seitens der Entwickler reagiert wurde. Der allgemeine Eindruck auf die Anwender schien äußerst positiv. Selbst als ein Volkskammer-Mitglied die Argumente zitierte, die im Volkskammerbereich zu Streitereien geführt hatten, wurden diese im öffentlichen Forum abgewehrt, die Diskussion verlief weiter ruhig. Grund dieser unterschiedlichen Reaktionen könnte sein, dass die in die Entwicklung einbezogenen Anwender sich mehr eigene Vorteile davon versprochen hatten, dass sie selbst Zeit und Mühe investiert hatten.

Phasenweise stellte die programmiertechnische Unwissenheit der meisten Anwender ein Problem dar, da diese nicht wussten, wie ein Software-Entwicklungsprozess vor sich geht. Vor allem die Trennung zwischen Design und Bedienkonzept war dabei ein Problempunkt, da diese vielen Anwendern nicht bewusst war. Des Weiteren sei es schwierig gewesen, *„dass (...) die Leute wenig Erfahrung haben, wie man Produkte entwickelt, wie man das Ganze strukturiert und auch Arbeitspakete separiert, und dann geht es auch manchmal drunter und drüber“* [Anhang – Interview 25]. Andererseits jedoch bringe die Denkweise der Nutzer auch überraschende Ideen hervor: *„Da hat einer so was Unreifes in den Raum gestellt und die nächsten drei haben das dann ein bisschen fein geschliffen und irgendwann war es dann richtig gut als Idee“* [Anhang – Interview 25].

Die späte Veröffentlichung der Alpha-Versionen war das Resultat von Problemen in der Zusammenarbeit Omegas mit einem externen Entwickler-Team. Zeitliche Verzögerungen stellten ein großes Problem dar, trotz fester Fristen ließen sich die Entwickler nicht unter Druck setzen. Auch in der Festlegung von Prioritäten hielten sich die Programmierer nicht an die Anweisungen des Omega-Teams, was dazu führte, dass die frühen Versionen einige weniger wichtige Features enthielten, es jedoch an Grundlagen fehlte. Selbst in der ersten Beta-Version fehlten lange versprochene Basis-Funktionen, was erneut zu Frustration bei den Anwendern führte. Auch das Kommunikationsverhalten der Externen sei mangelhaft gewesen,

sie waren weder telefonisch noch per E-Mail gut zu erreichen, Fehlermeldungen wurden von ihnen wie persönliche Angriffe aufgefasst. Die meisten Beschwerden der Omega-Mitarbeiter betrafen die Rolle des Projektleiters, die einer der externen Entwickler übernehmen sollte. Dieser beschränkte jedoch seine Aufgaben auf den Bereich der Programmierung. *„Das ist das allergrößte Problem, (denn er) hätte zu einigen Punkten auch am meisten dazu sagen können. (...) Da braucht man ja nur einen Satz zu schreiben, damit dann die Leute eigentlich schon mal zumindest ein Feedback bekommen“* [Anhang – Interview 14]. Da er dies jedoch genauso vernachlässigte wie seine weiteren Pflichten als Projektleiter, den Verlauf zu koordinieren, Zeitpläne zu erstellen und Meilensteine zu setzen, mussten diese Aufgaben wiederum von Omega-Mitarbeitern übernommen werden. Bei einigen dieser Mitarbeiter stellte es ein weiteres Problem dar, dass es ihnen an Motivation fehlte. Dies kritisierte der Produktmanager: *„Wir haben gesagt, wir machen das, und dann muss man auch an einem Strick ziehen und das ist nicht passiert. Aber (...) das hat nichts mit dem Konzept (Community) Driven Development zu tun, sondern schlichtweg mit der Einstellung, wie ich an das Thema ran gehe“* [Anhang – Interview 17].

6.7 Beurteilung des CDD-Ansatzes

Im Fokus dieses Kapitels steht die abschließende Bewertung des CDD-Ansatzes durch einerseits die am Prozess beteiligten Nutzer, andererseits die Meinung aller Anwender zum Gesamtergebnis. Zur Beurteilung des Konzepts wurde zum Einen ein quantitatives Feedback mittels eines selbst entwickelten Plug-Ins eingeholt und zum Anderen eine qualitative Begleitstudie durchgeführt. Diese Untersuchung bestand aus Gesprächen mit Nutzern und Entwicklern sowie einer Analyse der Forumdiskussion.

6.7.1 Quantitative Bewertung des Gesamtergebnisses

Um auch die Meinungen der Anwender zu erfassen, die nicht aktiv an der Entwicklung beteiligt waren, wurde in Absprache mit dem Omega-Team entschieden, eine Feedbackmöglichkeit in die Beta-Version der Software zu integrieren. Diese Idee entwickelte sich aus dem in Kapitel 6.5 vorgestellten Ansatz zur Einbindung der

Diskussion in das Produkt. Aufgrund der eingeschränkten Feedback-Möglichkeit über eine Fernbedienung kam es zu der Entscheidung, eine stark eingeschränkte Form des Feedbacks zu realisieren. Mittels einer skalenbasierten Notenvergabe sollten Nutzer die Möglichkeit erhalten, das System unterschieden nach Modulen sowie nach Aspekten bezüglich Funktionalität, Bedienbarkeit und Design zu bewerten. Die Abgabe einer einfachen freiwilligen Bewertung erschien der Zielvorgabe einer möglichst hohen quantitativen Teilnahme angemessen.

Nach einer gewissen Nutzungsdauer, dem jeweils fünften Start eines Programmteils, erschien ein Pop-up-Fenster, welches den Anwender zu einer Benotung des entsprechenden Moduls aufforderte. So gab es für die Kategorien „Funktionalität“, „Design“ und „Bedienbarkeit“ jeweils eine Bewertung von „sehr gut“ über „okay“ oder „ausreichend“ bis hin zu „schlecht“ zu vergeben. Die Skalen sollten ein differenziertes Bild über die Zufriedenheit der Nutzer mit dem jeweiligen Bewertungspunkt widerspiegeln. Durch eine automatische Auswertung der serverseitig gespeicherten Daten konnten die Entwickler somit erfahren, wie zufrieden die Anwender mit der Umsetzung ihrer Anforderungen waren und gegebenenfalls ebenfalls daran erkennen, in welchen Bereichen noch Veränderungen eingebracht werden sollten.

Realisierung

Der Prototyp wurde in Form eines C#-Plug-Ins entwickelt, welches auf einer vom Omega-Team angebotenen Schnittstelle auf COM-Basis aufsetzte. Dieses Interface ermöglicht die Anbindung von Plug-Ins in den Sprachen C#, Delphi und Visual Basic. Die übertragenen Informationen wurden über eine PHP-Datei in der serverseitigen Datenbank gespeichert, die anschließende Auswertung fand ebenfalls mittels PHP statt. Dafür wurde unter Zuhilfenahme von [Felk05] ein PHP-Skript entwickelt, das die gespeicherten Bewertungen zu anschaulichen Diagrammen formte. Der Quellcode sowohl des Plug-Ins als auch der PHP-Dateien ist im Anhang zu finden.

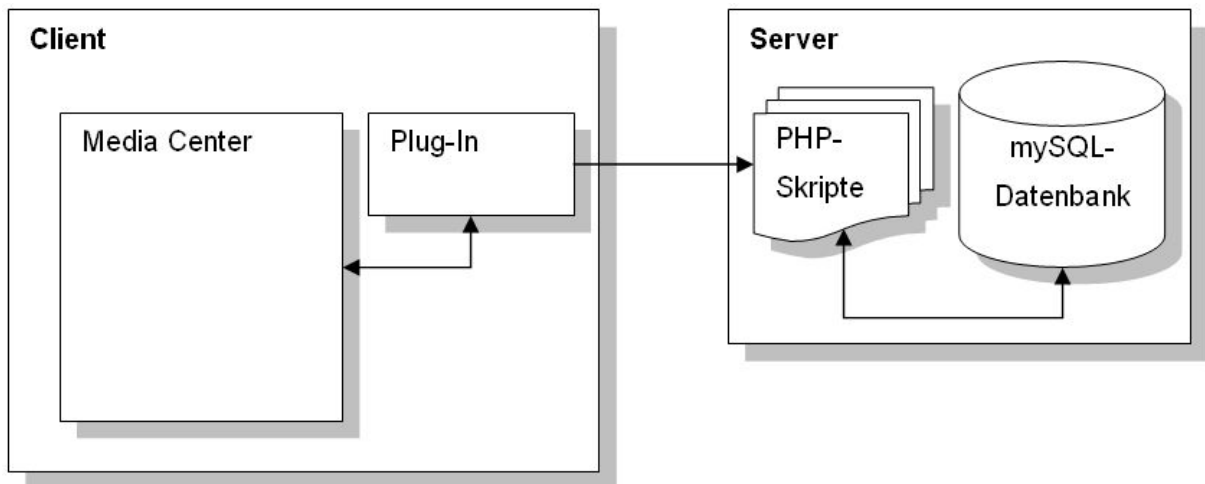


Abbildung 9: Client-Server-Architektur der Plug-In-Anbindung

Zur Entwicklung des Plug-Ins wurde die kostenlose Express Edition von Microsoft Visual C#¹¹ genutzt. Des Weiteren diente für Tests eine lokale mySQL-Datenbank, die mittels der ebenfalls kostenlosen Software XAMPP Lite¹² verwaltet wurde. Für die Speicherung der Ergebnisse in der Praxis stellte Omega einen Server bereit.

Ergebnisse

Die Möglichkeit der Feedback-Abgabe wurde von vielen Anwendern positiv aufgenommen. Auch viele Nutzer, die nicht aktiv an der Software-Entwicklung beteiligt waren, waren auf diese Weise bereit, ihre Meinung an die Entwickler zu übermitteln. In der siebenwöchigen Laufzeit gaben 707 Personen eine Bewertung für das TV-Modul ab. Dieses war das meistbewertete Modul, da es von nahezu allen Anwendern genutzt wird. Im Gegensatz dazu bewerteten das Bilder-Modul lediglich 35 Personen.

Da die Alpha-Versionen in ihrem Funktionsumfang noch sehr eingeschränkt waren, wurde entschieden, das Plug-In erst mit der ersten Beta-Version zu veröffentlichen, die alle Module mit ihren Grundfunktionen enthielt. Bewertet werden konnten somit die Module *TV*, *Musik*, *Video*, *EPG* und *Bilder*. Aufgrund der eher negativen

¹¹ <http://www.microsoft.com/germany/msdn/vstudio/products/express/download.msp>

¹² <http://www.apachefriends.org/de/xampp-windows.html>

Reaktionen der Volkskammer-Mitglieder auf die erste Alpha-Version¹³ und der zunehmenden Zurückhaltung bei Veröffentlichung der nachfolgenden Versionen, befürchteten die Entwickler zunächst, dass der Großteil der Anwender unzufrieden sein könne. Die Ergebnisse der gesammelten Bewertungen zeigten jedoch einen überwiegend positiven Gesamteindruck. Design, Funktionalität und Bedienbarkeit der einzelnen Module wurden von durchschnittlich zwei Drittel der Anwender mit „sehr gut“ oder „okay“ bewertet. Auch wenn die Ergebnisse nur eine erste Tendenz widerspiegeln können, so scheint doch die Mehrheit der Nutzer mit den Ergebnissen des CDD zufrieden zu sein.

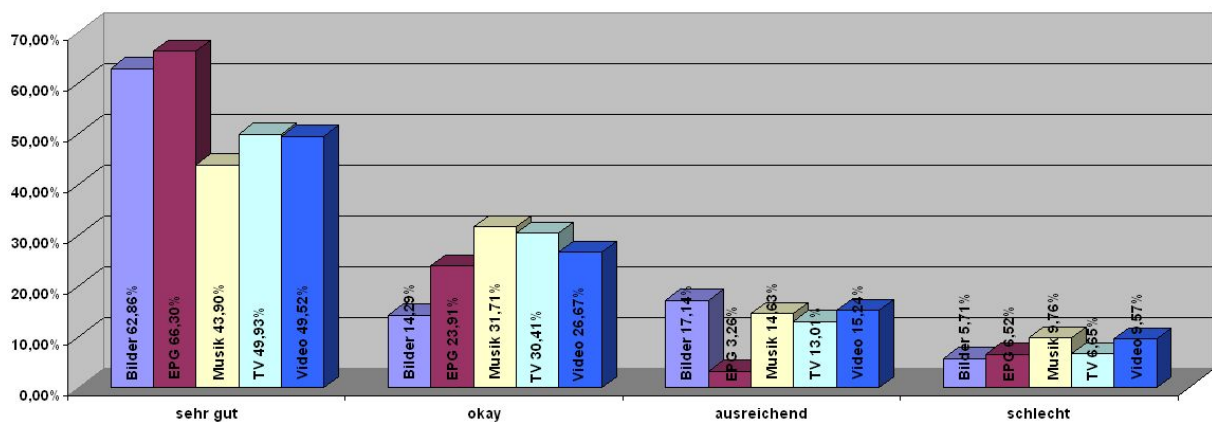


Abbildung 10: Bewertung des Designs (Stand vom 21.01.2008)

In Bezug auf das Design erweist sich vor allem das EPG-Modul als äußerst beliebt, die Komponenten Musik und Video hingegen erscheinen vergleichsweise negativ aufzufallen.

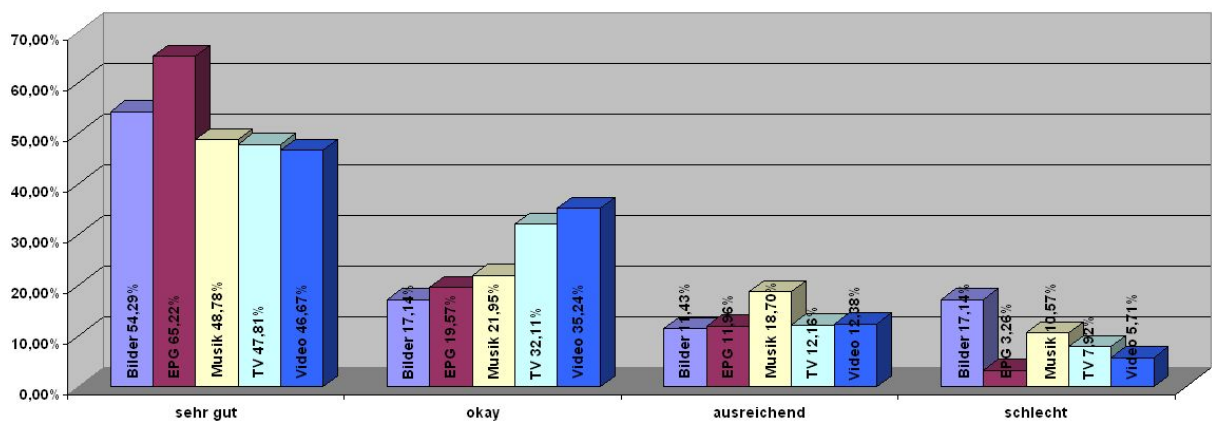


Abbildung 11: Bewertung der Funktionalität (Stand vom 21.01.2008)

¹³ Siehe Kapitel 6.6

Auch bezüglich der Funktionalität wird der EPG überdurchschnittlich positiv aufgenommen, das Bilder-Modul jedoch scheint im Vergleich zu den anderen Komponenten viel Verbesserungspotential zu bieten.

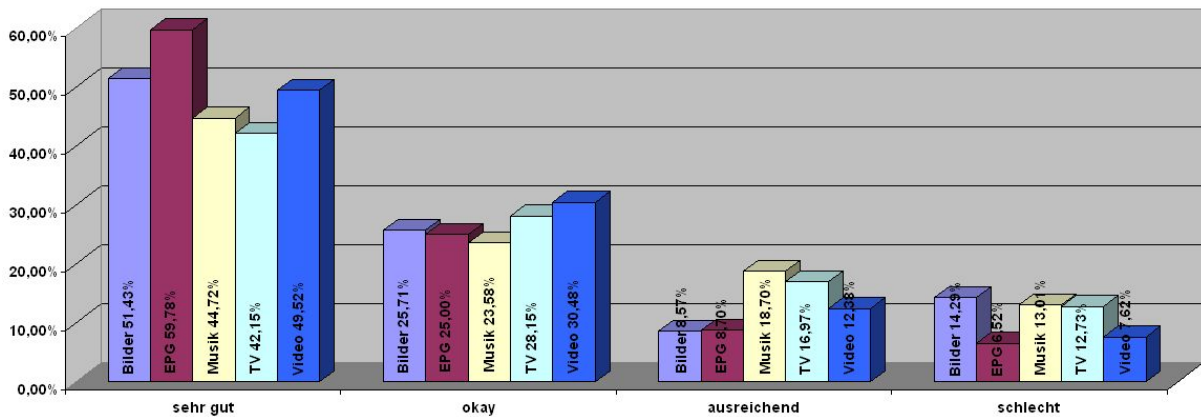


Abbildung 12: Bewertung der Bedienbarkeit (Stand vom 21.01.2008)

Als beste Komponente des Systems erweist sich der EPG auch in Bezug auf seine Bedienbarkeit. Musik- und TV-Modul hingegen erscheinen vergleichsweise schlecht bedienbar.

Auch wenn die Methode des quantitativen Feedbacks nur sehr eingeschränkte Rückschlüsse auf die Zufriedenheit der Anwender zulässt, so kann man feststellen, dass ein Großteil der Nutzer mit dem Ergebnis insgesamt zufrieden ist. Das nach Modulen und den drei Aspekten Funktionalität, Design und Bedienbarkeit kategorisierte Feedback erlaubt es den Entwicklern, die schlechter bewerteten Module in Bezug auf die jeweiligen Aspekte erneut hinsichtlich möglicher Verbesserungen diskutieren zu lassen und darauf aufbauend zu überarbeiten. Im betrachteten Anwendungsfall sollten somit insbesondere die Funktionalität des Bilder-Moduls sowie die Bedienbarkeit der Komponenten Bilder, Musik und TV erneut untersucht werden.

6.7.2 Qualitative Beurteilung des Ansatzes

Die qualitative Analyse des Konzepts beruht auf Untersuchungen der Forumdiskussion sowie den Ergebnissen der Interviews mit Entwicklern und Anwendern. Obwohl nahezu alle befragten Personen den Ansatz des CDD als positiv bewerteten, herrschte in der Volkskammer zeitweise große Unzufriedenheit. Die ersten Alpha-Versionen der Software trafen nicht die Erwartungen der einbezogenen Anwender.

Innerhalb der Volkskammer reagierten viele Nutzer unzufrieden, da sie ihre Vorstellungen nicht hatten durchsetzen können. Als seitens der Firma schnell auf den Unmut dieser Anwender reagiert wurde, beruhigte dies zwar die Diskussion, die Volkskammer-Mitglieder blieben jedoch weiterhin kritisch, was sich auch in den Interviews bestätigte. Externe Nutzer hingegen reagierten sofort äußerst positiv und zufrieden auf die Alpha-Versionen. Somit zeigt sich, dass das CDD zwar nicht alle Anwender komplett zufriedenstellt, jedoch für den Großteil der Nutzer gute Ergebnisse liefern kann. Schwierigkeiten waren vor allem auf mangelnde oder missverständene Kommunikation zurückzuführen. Eine verbesserte Unterstützung in diesem Bereich scheint dringend nötig. Damit könnten die Anwender ihre Ideen und Meinungen leichter in die Entwicklung einbringen. Außerdem könnten Missverständnisse und somit Enttäuschungen der am CDD beteiligten Nutzer vermieden werden.

6.7.3 Probleme und Verbesserungspotentiale

Die Erfahrungen aus dem Anwendungsfall zeigen, dass besonders in den Bereichen der Kommunikation und der zeitlichen Abstimmung Schwierigkeiten auftraten. *„Entweder man kommuniziert oder gibt den Leuten was, worüber sie selber schreiben können. Das ist halt beides nicht in dem Maße passiert, wie es hätte sein sollen“* [Anhang – Interview 17]. Zwar besserte sich das Informationsverhalten des Omega-Teams im Verlauf des Projekts, jedoch besteht weiterhin viel Verbesserungspotential in diesem Bereich. Auch die Kommunikation zwischen Zentralrat und Volkskammer empfanden viele Mitwirkende als gestört. Die Volkskammer warf dem Zentralrat vor, zu wenige Informationen an sie weitergeleitet zu haben, der Zentralrat hingegen beschwerte sich darüber, dass die Einträge im Wiki-System kaum gelesen wurden. Eine einfache Möglichkeit, um diese Lücke in der Kommunikation zu beheben, wäre es, in den entsprechenden Threads im Forum darauf hinzuweisen, sobald Informationen zu dem jeweiligen Bereich im Wiki-System erscheinen. So müssten die Mitglieder des Zentralrats die Beiträge nicht doppelt veröffentlichen, die Volkskammer wüsste jedoch, wann es Neuerungen gibt, und müsste somit nicht immer wieder das komplette Pflichtenheft danach durchsuchen. *„Hätte man sich die Zeit genommen – ich sage nicht, hätte man die Zeit gehabt, sondern hätte man sich die Zeit genommen – hätte man über den ganzen Projektverlauf die User begeistern können und mit involvieren können“* [Anhang – Interview 17].

Der Bereich der Kommunikation war im untersuchten Unternehmen schon oft als schwierig und vor allem seitens der Entwickler vernachlässigt aufgefallen. Somit ist anzunehmen, dass die entsprechenden Probleme in diesem Fall zusätzlich verstärkt waren. Jedoch ist die Kommunikation bei jedem CDD-Projekt von zentraler Bedeutung und sollte deshalb besonders gefördert werden. Es empfiehlt sich daher eine Person aus dem Entwickler-Team oder dem Zentralrat zu bestimmen, die für die tägliche Verbreitung aktueller Informationen und Fortschritte zuständig ist. So erläuterte ein Omega-Mitarbeiter: *„Wenn man sagt, man will User Driven Development machen, dann muss man kommunizieren, viele Informationen einholen und auch wieder zurückgeben. Da hätten wir fast einen Fulltime abstellen müssen, dann hätte so was wirklich auch sinnvoll funktionieren können“* [Anhang – Interview 16].

Auch die zeitlichen Probleme Omegas sind bei der Übertragung auf allgemeine Anwendungsfälle zu beachten. Dass das Projekt stark marktwirtschaftlich getrieben war, sei *„das Allerschlimmste, was (beim CDD) passieren kann“* [Anhang – Interview 14]. Termine und Fristen sollten nicht im Vordergrund stehen, existiert jedoch ein Zeitplan, so sollte dieser allen Beteiligten verdeutlicht werden. Die Dauer eines CDD-Projekts ist im Voraus meist schwer einzuschätzen, da die Anwender die Zeit, die sie in das Projekt investieren, selbst bestimmen und einteilen. Nahezu alle Beteiligten in der Fallstudie betrachteten den von Omega vorgesehen Zeitraum als zu knapp bemessen. Als für dieses Projekt angemessen schätzten sie durchschnittlich eine Dauer von mindestens einem Jahr. Einer der Entwickler erläuterte im Interview, für das CDD sei etwa die anderthalbfache Dauer der entsprechenden Entwicklung ohne Nutzerbeteiligung anzurechnen. Dies sei besonders dafür wichtig, einen iterativen Prozess zu ermöglichen, der die Vorteile der Nutzereinbeziehung zum Tragen kommen ließe. Dazu sei es allerdings auch zwingend notwendig, den Anwendern früh genug Zwischenergebnisse wie Prototypen zu liefern. Diese sollen einerseits dazu dienen, den Entwicklungsfortschritt sowie das Verständnis der Entwickler für die Vorschläge der Anwender zu überprüfen, andererseits dazu, die Volkskammer-Mitglieder zu neuen Ideen anzuregen, da ein sehr abstraktes Denken für viele Anwender ein Problem darstelle. Auch für die Planung im Vorfeld eines solchen Projekts sollte genügend Zeit vorgesehen werden. *„Wir haben zum Teil gar keine*

richtigen Richtlinien vorgeben können, weil wir gar nicht gewusst haben, wie kriegen wir denn die Leute dazu, überhaupt vernünftig Informationen - über welche Weg überhaupt - (zu liefern)“ [Anhang – Interview 14]. Eine genauere Planung im Voraus hätte zu mehr Klarheit über die Vorgehensweise führen können. Ebenso hätte ein gut durchgeführtes Projektmanagement im Verlauf des Projekts für eine bessere Struktur gesorgt.

Der Zentralrat

Im Projektverlauf kam es innerhalb des Zentralrats immer häufiger zu Schuldzuweisungen zwischen Anwendern und Omega-Mitarbeitern. Vor allem bezogen sich diese auf die Verteilung der Aufgaben innerhalb des Gremiums. Das Team fühlte sich teilweise von den Nutzern im Stich gelassen, während die Anwender über mangelnden Einfluss und fehlende Informationen klagten. Eine klare Aufgabenteilung zu Beginn des Projekts hätte die Zusammenarbeit voraussichtlich stark vereinfachen können. Viele Teilnehmer wünschten sich im Nachhinein die Festlegung einer Person als Kommunikationsbeauftragten des Zentralrats. Dessen Aufgabe sei es, täglich den Fortschritt des Projektverlaufs sowie neue Ergebnisse zwischen den unterschiedlichen Gremien zu vermitteln. Diese Tätigkeit sollte möglichst ein Mitarbeiter des Omega-Teams übernehmen, der nah am internen Geschehen beteiligt sei [vgl. Anhang – Interview 16]. Durch die knappen Personalressourcen war es Omega nicht möglich, einen Mitarbeiter in solchem Ausmaß mit diesen Pflichten zu beauftragen. Für zukünftige Projekte sollte eine solche Lösung jedoch angedacht werden.

Ein weiteres Problem war die Besetzung des Zentralrats. Die meisten Beteiligten empfanden diesen als zu klein und vor allem von der Nutzerseite zu gering besetzt. Durch den krankheitsbedingten Ausfall eines Anwenders ging schließlich auch die Mehrheit der Nutzer im Zentralrat verloren. Als es in der zweiten Projektphase zu Streitigkeiten über die weitere Vorgehensweise kam, trat ein weiterer Anwender aus dem Gremium aus, so dass letztlich die drei Entwickler den verbliebenen beiden Nutzern gegenüberstanden. Auch die personelle Besetzung des Zentralrats schien nicht optimal gewählt. So beteiligte sich einer der Entwickler kaum an den Diskussionen und Telefonkonferenzen, obwohl dieser als Programmierer und

Projektleiter an vielen Punkten entscheidende Informationen, beispielsweise über die generelle Umsetzbarkeit von Ideen, hätte geben können. Seitens der Anwender war es ein Problem, dass diese phasenweise durch berufliche und private Verpflichtungen wenig Zeit für die Zusammenarbeit zur Verfügung hatten. Bei einem größeren Gremium wären einzelne Ausfälle nicht von solcher Bedeutung gewesen. Ein weiterer schwieriger Aspekt war die tatsächliche Motivation, da der Zentralrat ein neutrales Gremium sein sollte. Einige Beteiligte vermuteten jedoch, manche der einbezogenen Anwender hätten versucht, ihre eigenen Interessen in den Vordergrund zu stellen. Es soll sogar vorgekommen sein, dass jemand seine Ideen selbst ohne die Zustimmung des Zentralrats durchzusetzen versuchte. Ein Omega-Mitarbeiter hingegen erklärte, dass es auch dem Entwickler-Team leicht fiel, eigene Wünsche durchzubringen: *„Vielleicht haben wir bewusst oder unterbewusst schon mal Leute gewählt (...), die so tendenziell auf unserer Seite sind, die man schnell von was überzeugen kann und die da vielleicht auch nicht mehr so die Meinung nach außen hin, ihre eigene Meinung, vertreten“* [vgl. Anhang – Interview 14]. Um solchen Problemen entgegen zu wirken, äußerte ein anderer Mitarbeiter die Idee, die Mitglieder des Zentralrats im Laufe des Projekts auszutauschen, um Stagnation zu vermeiden. Auch der organisatorische Ablauf der Zusammenarbeit des Zentralrats bereitete Schwierigkeiten. Terminliche Abstimmungen für Telefonkonferenzen und Treffen waren oft kompliziert, so kam es letztlich zu nur zwei persönlichen Treffen im gesamten Projektverlauf. In den Telefonkonferenzen kamen trotz meist etwa zweistündiger Dauer einige Personen kaum zu Wort. Aus diesem Grund sprachen sich die meisten Beteiligten dafür aus, bei einem zukünftigen Projekt mehr persönliche Treffen, möglichst monatlich, zu veranstalten.

Omega-Mitarbeiter äußerten Bedenken, dass vor allem die Anwender im Zentralrat die ihnen verliehene Macht missbrauchen könnten. Dies könne geschehen, indem die Nutzer ihre Zustimmung zur Veröffentlichung der Software verweigern, solange nicht all ihre persönlichen Wünsche erfüllt seien. Im Fall Omega sei die fehlende Zustimmung mancher Zentralrats-Mitglieder zwar darauf begründet, dass das System noch nicht so weit entwickelt war, wie es die ursprüngliche Planung vorgesehen hatte, die Gefahr des Machtmissbrauchs sei jedoch generell vorhanden [vgl. Anhang – Interview 17]. Diese Angst der Entwickler widerspricht allerdings dem

grundsätzlichen Gedanken des CDD, denn dieses baut vor allem auf dem Vertrauen der Entwickler in die Arbeit der Anwender auf. *„Wenn ich sage, `User Driven Development, ich habe ein Zentralkomitee, was die Entwicklung steuert`, dann muss ich den Leuten auch vertrauen“* [Anhang – Interview 19]. Neben der Entscheidungsgewalt betreffe dies auch den Informationsfluss, die einbezogenen Nutzer sollten den anderen Zentralrats-Mitgliedern gleichgestellt sein und die gleichen Informationen erhalten wie diese. Gerade weil der Großteil der Entwickler die Software selbst kaum nutze, sei es wichtig, den Anwendern zu vertrauen und das CDD nicht nur als *„Marketing-Schlachtwort“* zu nutzen, sondern tatsächlich anzuwenden, *„einfach zu sagen, `okay, wir machen mal`, weil das passiert natürlich bei einem Apple oder Microsoft nicht, dass dort gesagt wird, `okay, wir gehen jetzt direkt auf das ein, was ihr wünscht, wir stellen das um`. Aber das ist bei Omega auch nicht passiert“* [Anhang – Interview 19].

Die Volkskammer

Obwohl die meisten Entwickler betonten, dass die Arbeit der Volkskammer viele interessante Ideen hervorgebracht habe, wurde die Beteiligung innerhalb dieses Gremiums als zu gering eingeschätzt. Von den ursprünglich 70 Teilnehmern beteiligten sich nur etwa 15 Personen regelmäßig. Nach Einschätzung der Omega-Mitarbeiter seien dies zu wenig Anwender gewesen. *„Wenn man da ein bisschen mehr Leute gehabt hätte, dann wäre vielleicht die Diskussion auch reger gewesen, dann hätten andere Leute vielleicht auch nicht so schnell das Interesse verloren“* [Anhang – Interview 14]. Die in der ursprünglichen Planung vorgesehenen 70 Anwender seien dafür eine gute Anzahl. Neben dem Vorteil, dass mehr Beteiligte eine größere Zahl an Ideen hätten liefern können, hätten mehr Nutzermeinungen auch einen genaueren Hinweis darauf geben können, ob die Ideen für die Zielgruppe oder nur für Einzelne interessant seien. *„(Je) kleiner die Nutzerzahl, User sagen, was sie wollen, desto schlechter kann man abschätzen und abwägen, ob das wirklich sinnvoll ist oder nicht. Und dann fließen auch zu viele eigene Argumente mit rein.“* So sei durch die geringe Beteiligung innerhalb der Volkskammer zu wenig Druck auf die Zentralrats-Mitglieder entstanden, in diesen Punkten den öffentlichen Meinungen

nachzugeben. Dadurch seien eventuell verstärkt die Interessen der einzelnen Zentralrats-Mitglieder eingeflossen.

Fazit der Beteiligten zum Konzept des CDD

Zu Beginn des Projekts war die Stimmung unter den Anwendern optimistisch, wobei sie keine allzu hohen Erwartungen an das Ergebnis hatten. Allein die Möglichkeit, sich an der Entwicklung beteiligen und eigene Ideen einbringen zu können, schätzten die Nutzer als sehr wertvoll. Seitens der Entwickler herrschte deutlich mehr Skepsis. Zwar sahen sie die Chancen des CDD, jedoch auch viele Risiken. Obwohl sich im Verlauf des Projekts einige der Befürchtungen bestätigten, wurde das CDD-Konzept zum Ende von beiden Gruppen als positiv eingestuft. Zwar sei das Ergebnis des CDD im Anwendungsfall kein Erfolg, jedoch sei es *„ein größerer Erfolg (...), dass die Methode an sich offensichtlich funktionieren kann und vielleicht bei anderen Projekten (...) deutlich besser funktioniert als Methode als es hier in dieser speziellen Konstellation war“* [Anhang –Interview 18]. Bis auf eine Person gaben alle Entwickler an, sie würden erneut ein CDD anwenden, dann jedoch die entdeckten Problemfelder beseitigen. Vor allem sei es ihnen wichtig, mehr Zeit für das Projekt einzuräumen. Seitens der Anwender herrschte zwar weniger Bereitschaft, in Kürze erneut an einem solchen Projekt teilzunehmen, jedoch begründeten sie dies mit persönlichen Verpflichtungen, das Konzept habe sie hingegen überzeugt. *„User Driven Development funktioniert, wenn (...) bestimmte Sachen dann auch klar definiert sind, die Aufgaben klar verteilt sind, es vom Team mitgetragen wird, mit den Usern auch kommuniziert wird - dann funktioniert es“* [Anhang – Interview 22]. Unter diesen Voraussetzungen sei die Methode dazu geeignet, ein aktuelles Problem des Marktes zu lösen. Seit etwa 15 Jahren werde Software kaum noch über neue Funktionen verkauft, sondern immer stärker über die Bindung der Kunden an das Produkt. *„Um diese Anwender in so ein Commitment zu einem Produkt zu bringen, ist es natürlich eine sehr geeignete Methode“* [Anhang – Interview 18].

7 Diskussion

Die in der Praxis-Studie gewonnenen Erkenntnisse werden in diesem Kapitel entsprechend den in 5.3 vorgestellten Forschungsfragen genauer analysiert und beschrieben. Dabei wird ebenfalls untersucht, inwiefern aufgetretene Schwierigkeiten und Ergebnisse unternehmens- oder projektspezifisch waren oder sich auf allgemeine Anwendungsfälle übertragen lassen.

7.1 Motivation

Wie ist das Selbstverständnis der Anwender für ihre Nutzer-Community?

In den Interviews erklärten die meisten Anwender, für ein Gemeinschaftsgefühl seien ihnen vor allem Spaß und ein friedliches Miteinander von Bedeutung. Teilweise stelle dies im Omega-Forum ein Problem dar, da es vom Großteil der Anwender nur genutzt werde, um Hilfe bei Schwierigkeiten mit der Omega-Software zu erfragen. *„Da ist alles so auf Kontra. (...) Sind halt zu 80% nur frustrierte User“* [Anhang – Interview 09], beschrieb dies ein Anwender. Jedoch lenkte auch dieser im Nachhinein ein, dass in einer Kerngruppe von etwa 15 bis 20 Personen ein Gemeinschaftsgefühl herrsche. Diese Hauptgemeinschaft, deren Größe in der Beschreibung anderer Anwender meist noch geringer ausfiel, bestünde fast ausschließlich aus Moderatoren und habe sich durch die Gespräche im Moderatorenforum, vor allem jedoch über Moderatorentreffen gebildet. Ein weiterer Anwender erklärte, das Gemeinschaftsgefühl herrsche zwar nur im engeren Kreis mit den Personen, mit denen permanent aktiver Kontakt bestünde, zur Community gehören jedoch alle Nutzer. Wiederum ein anderer Nutzer definierte zwei unterschiedliche Gemeinschaftsgefühle im Forum, die der Pro- und der Contra-Gruppe. Er selbst zähle sich zur positiv denkenden Gruppe, die im Forum die Vorteile der Produkte vertrete und Hilfestellungen gebe. Im Gegensatz dazu stehe die negativ eingestellte Contra-Gruppe, die sich über kleinste Systemfehler aufrege und meist Streit suche [vgl. Anhang – Interview 13]. Eine nochmals andere Ansicht hatte ein Anwender, der Meinungsführerschaften und eine Hierarchie innerhalb des Forums identifizierte. *„Es gibt halt ein paar, die stechen heraus und nehmen sich auch das ein oder andere gegenüber anderen Usern heraus“* [Anhang – Interview

09]; dies seien zum Großteil Moderatoren und Beta-Tester. Weitere Hierarchiestufen bilden die gewöhnlichen Anwender je nach Grad ihrer aktiven Beteiligung im Forum. Ein interessanter Aspekt war, dass sich bis auf einen Anwender jeder der befragten Nutzer zur besagten Kern-Community zählte.

Was motiviert Nutzer, am Designprozess zu partizipieren?

Die meisten Anwender begründeten ihre Teilnahme am Projekt mit der Möglichkeit, ihre eigenen Ansichten und Ideen in die Entwicklungsarbeit einzubringen. Manche Nutzer bezeichneten dies als Egoismus, andere wiederum als eine Hilfestellung für das Entwickler-Team, was widerspiegelt, dass die Mitwirkung der Anwender Vorteile für beide Seiten mit sich bringt. Die Erfahrungen der Nutzer aus Alltagssituationen können neue Perspektiven liefern, die den Entwicklern bisher nicht geläufig waren. Manche Anwender hatten schon zu Beginn konkrete Vorstellungen, welche speziellen Systemfunktionen sie durch ihre Beteiligung durchsetzen wollten, anderen waren generelle Faktoren wie Stabilität und Bedienbarkeit wichtiger. Der Spaß an der Zusammenarbeit sowohl mit dem Entwickler-Team als auch mit anderen Nutzern motivierte wiederum andere Anwender. Finanzielle Anreize gab es in diesem Anwendungsfall nicht, die Nutzer erklärten sich freiwillig zur Mitwirkung bereit. Der wichtigste Motivationsaspekt für sie schien dabei die Einflussnahme auf die Entwicklung eines optimal auf sie angepassten Produktes zu sein.

Somit entsprechen die in dieser Studie identifizierten Motivationsgründe zur Teilnahme an einem CDD-Konzept den bereits vorgestellten Anreizen zur Beteiligung an herkömmlichen PD-Projekten¹⁴. Der stärkste Motivationsaspekt scheint die Hoffnung auf ein verbessertes, den eigenen Wünschen entsprechendes Produkt zu sein. Weiterhin sind Neugierde und Spaß entscheidende Faktoren.

Was motiviert die Designer, Nutzer stärker in diesen Prozess einzubeziehen?

Nach Meinung sowohl der Anwender als auch der Entwickler bieten die Omega-Produkte eine technisch gute Basis. In den letzten Jahren kam es jedoch häufiger vor, dass an der Zielgruppe vorbei entwickelt wurde. Einige Features vorheriger

¹⁴ Siehe Kapitel 3.4 und 4.2

Versionen wurden kaum genutzt, weil sie nicht den alltäglichen Bedürfnissen der Anwender entsprachen. Um diesem Problem entgegenzuwirken schien eine direkte Einbeziehung der zukünftigen Anwender sinnvoll [vgl. Anhang – Interview 01]. Somit ließen sich die wahren Bedürfnisse der Anwender ermitteln und neue Ideen aus der Anwenderperspektive sammeln. Geben die einbezogenen Nutzer eine für den Markt der zukünftigen Endanwender repräsentative Meinung wieder, so sei eine bessere Software zu erwarten, die durch ihre Ausrichtung auf die Kundenbedürfnisse Omega eine klare Alleinstellung im Markt verschaffen könne. Da es eine nahezu „*zwangsläufige Entwicklung*“ des Marktes sei, sei auch dem Omega-Team wichtig, „*dass man anfängt, den Anwender deutlich ernster zu nehmen und den Kunden als Anwender deutlich stärker in den Fokus zu stellen*“ [Anhang – Interview 02].

Einigen Entwicklern war jedoch eine große Skepsis und somit eine Ablehnung der Kundeneinbeziehung anzumerken. Dem *Not Invented Here Syndrom*¹⁵ sollte durch die Mitwirkung von Gatekeepern, den Mitgliedern des Zentralrats, entgegengewirkt werden, die das Vertrauen der Entwickler genießen. „*Es gibt, gerade unter den Moderatoren, sehr viele User, die uns schon sehr lange treu sind, die uns schon sehr lange begleiten, schon ein paar Jahre und länger, die auch mit uns durch Höhen und Tiefen gegangen sind, die sehr gut wissen, was wir tun, wie wir arbeiten, was unsere Software kann und was sie nicht kann*“ [Anhang – Interview 02]. Das durch langjährige Kontakte gestärkte Vertrauensverhältnis war somit eine optimale Voraussetzung für die Zusammenarbeit. Neben der Verbindung über das Forum bestanden zu einigen Moderatoren und Beta-Testern auch telefonische Kontakte. Gelegentliche Treffen haben zusätzlich bewirkt, dass ein nahezu freundschaftliches Verhältnis zwischen dem Entwickler-Team und diesen Personengruppen aufgebaut wurde. Somit schienen sich für die Besetzung des Zentralrats gerade Personen aus diesem Anwenderkreis zu eignen.

¹⁵ Siehe Kapitel 5.2

7.2 CDD-Unterstützung

Wie kann man den Austausch zwischen der Community und den Designern zukünftig geeignet unterstützen?

Nach der Analyse der ersten Projektphase zeigten sich die größten Schwierigkeiten in fehlender oder missverstandener Kommunikation zwischen den Gremien sowie in der zeitlichen Planung. Unterschiedliche Mechanismen zur Verbesserung der Kommunikation wurden bereits in Kapitel 7.2 präsentiert. Weitere Tools sowie in den Interviews ermittelte Ideen der Anwender und Entwickler werden im Folgenden vorgestellt.

Die Wahl der Kommunikationsmedien im untersuchten Anwendungsfall hielten einige Beteiligte für unvorteilhaft; dies betraf sowohl das Forum, das Wiki-System als auch die Telefonkonferenzen des Zentralrats. Vor Projektbeginn waren für die Arbeit der Volkskammer andere Möglichkeiten angedacht wie beispielsweise ein White Board oder ein graphisches Design-Tool. Durch eine zu kurze Vorlaufphase des Projekts ließ sich eine solche Variante jedoch nicht rechtzeitig umsetzen, weshalb auf die bestehenden Mittel zurückgegriffen wurde. Im Hinblick auf zukünftige Projekte empfahlen die Anwender in den Interviews eine stärkere Verbindung von Diskussion und Information, beispielsweise indem zu beiden Zwecken das Wiki-System genutzt werde. Von Entwicklerseite hingegen wurde eine noch stärkere Trennung dieser Bereiche gewünscht, zum Beispiel durch die Kombination eines Diskussionsforums mit einem Entwickler-Blog. Die Telefonkonferenzen des Zentralrats wurden ebenfalls als schlecht geeignetes Mittel angesehen, da meist zu etwa 80 Prozent der Zeit nur zwei Personen sprachen, die anderen kaum zu Wort kamen. Ursprünglich waren mehr persönliche Treffen des Zentralrats vorgesehen und auch von allen Teilnehmern gewünscht, jedoch scheiterte dies immer wieder an den Abstimmungen von Termin und Ort. Für zukünftige Projekte empfahlen jedoch nahezu alle Beteiligten regelmäßige Treffen. Vor allem in der zweiten Projektphase wuchs auch bei den Mitgliedern der Volkskammer der Wunsch nach einer Veränderung der Medien. *„Es gibt viele Strukturen, die eigentlich da sind, die einfach (...) nicht richtig genutzt werden“* [Anhang – Interview 24]. Omegas Beta-Tester verwenden zum

Sammeln von Fehlermeldungen ein sogenanntes *Mantis*¹⁶. Nach der ursprünglichen Planung war auch für die Testanwender von OmegaTV4 ein solches Programm zum Dokumentieren von Fehlermeldungen vorgesehen. Doch auch dieses kam nicht zum Einsatz, obwohl es den Prozess sowohl für die Entwickler als auch die Tester stark vereinfacht, beschleunigt und doppelte Einträge vermieden hätte. Um die Volkskammer-Mitglieder intensiver in Entscheidungen einzubeziehen und ihnen das Gefühl zu geben, mehr bewirken zu können, seien Abstimmungen innerhalb ihres Unterforums sinnvoll. Diese waren bei den Anwendern immer sehr beliebt und erhöhten deren Diskussionsbereitschaft.

Auch die Teilnehmer selbst hatten einige Ideen, wie man die Zusammenarbeit verbessern könne. Als wichtigsten Ansatz nannten alle Befragten eine Verbesserung des Kommunikationsverhaltens. Eine Möglichkeit in dieser Hinsicht sei ein Entwickler-Blog, in dem die Programmierer den aktuellen Entwicklungsstand veröffentlichen. Zu Beginn des Projekts schrieb der Zentralrat im Anschluss an die Telefonkonferenzen einen Bericht über die getroffenen Beschlüsse. Dieser Informationsfluss brach jedoch nach wenigen Wochen ab. Ein wöchentliches Feedback wäre vor allem für die Mitglieder der Volkskammer sehr interessant. Dies brauche nicht ausführlich sein, eine kurze Liste oder tabellarische Übersicht genüge den meisten Anwendern. Wie bereits zuvor erläutert¹⁷, scheint die Wahl eines Kommunikationsbeauftragten des Zentralrats äußerst sinnvoll, der zwischen den verschiedenen Gremien vermittelt. Für eine bessere Strukturierung empfahl der Produktmanager die Einbringung des zusätzlichen Organs „Unternehmen“. Dieses sollte klare Randbedingungen und gegebenenfalls Fristen definieren.

Die passende Besetzung und ideale Größe des Zentralrats erwiesen sich als strittige Themen. Wunsch sowohl der Anwender als auch mancher Entwickler war eine Vergrößerung dieses Gremiums, wobei dies vor allem durch einen größeren Anteil an einbezogenen Anwendern geschehen sollte, beispielsweise in einem Verhältnis von zwölf Anwendern gegenüber drei Entwicklern. Um das Vertrauen auf Seiten der

¹⁶ <http://www.mantisbt.org/>

¹⁷ Siehe Kapitel 6.7.3

Nutzer zu erhöhen, sei es sinnvoll, die Mitglieder des Zentralrats von der Volkskammer wählen zu lassen [vgl. Anhang – Interview 18]. Um einerseits Machtmissbrauch zu vermeiden, andererseits die Diskussion lebendig zu halten, sei außerdem anzudenken, die Mitglieder des Zentralrats zwischenzeitlich auszuwechseln [vgl. Anhang – Interview 16].

Da viele Anwender kein abstraktes Denken gewohnt sind, kann es ihnen schwer fallen, ohne eine graphische Repräsentation des Systems zu arbeiten. Bei der Erzeugung neuer Versionen einer Software kann auf bestehenden Grundlagen aufgebaut und schnell ein Prototyp des neuen Systems erzeugt werden. Bei der Neuentwicklung von Software ist dies schwieriger, weshalb visuelle Hilfswerkzeuge gefragt sind. Diese können verhindern, dass schon in frühen Entwicklungsphasen Fehler in ein System eingebaut werden, die durch Missverständnisse zwischen Entwicklern und Anwendern entstehen. Viele Unternehmen, die einen PD-Ansatz verfolgen, nutzen somit spezielle Tools, um den Entwicklungsprozess zu unterstützen. Die meisten dieser Hilfswerkzeuge dienen den Zielen, Kommunikation zu erleichtern und Kreativität anzuregen. Außerdem sollen die Entwickler entlastet werden, indem Anwendern die Möglichkeit gegeben wird, einen kompletten Trial-and-Error-Prozess, also die Entwicklung sowie anschließende Tests ihrer Ideen, vorzunehmen und somit iterativ selbst eine Lösung ihrer Probleme zu finden. Für die Entwicklung von Softwareartefakten scheinen vor allem interaktive Basteltools geeignet, wie zum Beispiel Drag-and-Drop-Tools, die eine Zusammenstellung unterschiedlicher Optionen für Funktionalitäten und Verarbeitung ermöglichen [vgl. Hipp02]. Ebenso für diesen Anwendungsbereich interessant erscheinen Toolkits, die bei Änderungen direkt deren Auswirkungen auf Performanz und Aussehen am virtuellen Prototyp veranschaulichen [vgl. FüBa+06]. Die Anwender bringen somit nicht nur Ideen ein, sondern haben die Möglichkeit, unterschiedliche Aspekte zu beachten, vollständige Konzepte zu erstellen und Alternativen abzuwägen. Bei Nutzung solcher Tools sollte jedoch darauf geachtet werden, diese geeignet mit dem Diskussionsbereich zu verknüpfen, um keinen Medienbruch entstehen zu lassen. Somit scheint für ein CDD beispielsweise ein Tool sinnvoll, das mittels Drag-and-Drop ein einfaches Erstellen von Design-Entwürfen ermöglicht, die anschließend allen Anwendern beziehungsweise den Mitgliedern der Volkskammer zur Ansicht,

Bewertung und gegebenenfalls Verfeinerung zur Verfügung stünden. Ein ähnliches Tool nutzte beispielsweise der Sportartikelhersteller Adidas-Salomon bei der in Kapitel 4.2 vorgestellten Fallstudie. Ebenfalls zum Einsatz kommen können kommunikationsunterstützende Hilfssysteme wie beispielsweise Programme, die Chats um graphische Elemente oder einen Datenaustausch erweitern. Zu diesem Zweck könnte ein Tool wie das in Kapitel 6.5 vorgestellte *ConcertChat* verwendet werden. Damit ließen sich Gruppendiskussionen und gemeinsame Brainstormings ermöglichen. Bei einer Forumdiskussion können die Strukturierung unterstützt und die Auswertung automatisiert werden, wie es zum Beispiel durch das Tool *I-DIAG*¹⁸ geschieht [vgl. AcSw+03].

7.3 CDD-Konzept

Welche Probleme und Vorteile bringt die Einbeziehung einer Nutzer-Community im Gegensatz zur Partizipation einzelner Endanwender?

Probleme: Als größter Problembereich im untersuchten Anwendungsfall stellte sich die Kommunikation heraus. Von Seiten der Entwickler wurde zu wenig Zeit in diese investiert und auch die Weiterleitung der Informationen vom Zentralrat an die Volkskammer war unzureichend. Zur Verbesserung der Zusammenarbeit sollte in zukünftigen Projekten ein Mitglied des Zentralrats als Kommunikationsbeauftragter ernannt werden. Diese Person hätte die Aufgabe, täglich Informationen zwischen den Gremien zu vermitteln. Auch die verwendeten Kommunikationsmedien erwiesen sich als nicht optimal für ein CDD. Die Anwender schienen abstrakte Ideen schwer nachvollziehen zu können. Ein graphisches Unterstützungssystem könnte hilfreich sein, um Missverständnissen und Enttäuschungen vorzubeugen.

Ziel des CDD ist es, die Anwender den Entwicklungsprozess vorantreiben zu lassen. Trotz dessen bedarf ein solches Projekt der Steuerung und Kontrolle durch einen Projektleiter. Klare Richtlinien über die Vorgehensweise und Aufgaben der beiden Gremien sollten im Voraus festgelegt und ihre Einhaltung im Projektverlauf geprüft werden. Des Weiteren ist es Aufgabe der Projektleitung, innerhalb der Entwicklung Meilensteine zu setzen. Jedoch sollten Termine und Fristen des Unternehmens eine

¹⁸ I-DIAG = Interactive **Di**genesis

eher untergeordnete Bedeutung haben und die Entwicklung nicht unnötig einengen. Eine flexible Zeitplanung ist beim CDD von Vorteil, da die Anwender freiwillig mitwirken und ihre Zeit selbst einteilen. Aufgrund dessen und der größeren Zahl an Ideen und Meinungen, sollte für ein CDD mindestens die anderthalbfache Dauer einer Entwicklung ohne Nutzereinbeziehung vorgesehen werden.

Von großer Bedeutung ist das Vertrauen aller Beteiligten ineinander und in die gemeinsame Arbeit. Vor allem bei der Auswahl der Zentralrats-Mitglieder sollte darauf geachtet werden, dass diese von sowohl den Anwendern als auch den Entwicklern akzeptiert werden. Da sie in zentraler Position als Vertreter der Community agieren, wäre es sinnvoll, diese Personen von der Volksskammer wählen zu lassen. Dies sollte in Abstimmung mit den Entwicklern geschehen. Im untersuchten Anwendungsfall waren diese jedoch mit der Wahl der Community-Vertreter unzufrieden, obwohl sie diese selbst bestimmt hatten. Es bestand die Befürchtung, die Mitglieder des Zentralrats könnten ihre Macht missbrauchen. Um dem entgegenzuwirken, könnte die Besetzung des Gremiums in regelmäßigen Abständen ausgetauscht werden. Dann jedoch sollte die Kommunikation zwischen allen Gremien zuvor optimiert werden, um sicherzustellen, dass die neuen Zentralrats-Mitglieder den jeweils aktuellen Stand der Entwicklung und Entschlussfassung kennen.

Vorteile: Grundsätzliche Bedingung für die Beteiligung in einer Community ist ein gemeinsames Interesse aller Mitglieder. Bei einer Nutzer-Community bezieht sich dieses auf das Produkt bzw. den Markt, um den herum sich die Community aufbaut. Zwar beteiligen sich manche Anwender nur an Diskussionen, wenn sie Probleme mit dem Produkt haben und Hilfestellungen oder Empfehlungen wünschen. Der Kern der Community jedoch besteht aus einer Menge an Nutzern, die regelmäßig an Diskussionen mitwirken und sich somit sehr ausführlich mit dem Produkt bzw. Markt beschäftigen. Gerade in virtuellen Nutzer-Communities finden sich somit viele Anwender, die sich äußerst gut mit den Produkten bzw. Märkten auskennen und innovative Ideen liefern können.

Eine bereits etablierte Community bietet weiterhin den Vorteil, dass sich viele Mitglieder – sowohl Nutzer als auch gegebenenfalls Angestellte des Anbieters, die zu Support-Zwecken im Forum aktiv sind – bereits seit längerer Zeit über Forumsdiskussionen kennen. Wie bereits beschrieben herrscht auch bei der Omega-Community ein Gemeinschaftsgefühl unter einigen Anwendern und Mitarbeitern. Gerade für Mitglieder, die sich häufig im Forum aufhalten, kann es somit ein Anreiz sein, mit den anderen Community-Mitgliedern zusammen an einem solchen Projekt zu arbeiten. Es ist anzunehmen, dass Mitglieder einer Nutzer-Community eher dazu bereit sind an Designprozessen mitzuwirken als Kunden, die keine weitere Verbindung zum Anbieter und anderen Anwendern haben.

Den größten Vorteil der Einbeziehung einer Nutzer-Community stellt die Möglichkeit der Anwender zur Diskussion dar. Die Nutzer können Ideen austauschen, gemeinsam diskutieren und weiterentwickeln. Dies führt zu besser durchdachten Ergebnissen als beispielsweise die Befragung einzelner Anwender. In den Ergebnissen der Fallstudie wird dies ebenfalls deutlich. Betrachtet man die quantitative Auswertung, so lässt sich eine überwiegend positive Resonanz der Anwender für alle Module und Bewertungskategorien erkennen. Mehr als Dreiviertel der Nutzer empfanden die Software als „sehr gut“ oder „okay“, lediglich etwa neun Prozent der Anwender hielten das System für „schlecht“. Somit ist der Anteil der tatsächlich unzufriedenen Nutzer relativ gering. Aufgrund der qualitativen Analyse sowohl der Interviews als auch der Forumsdiskussion ist zu vermuten, dass die enttäuschten Anwender zum Großteil Personen sind, die am Entwicklungsprozess beteiligt waren und ihre eigenen Vorstellungen nicht durchsetzen konnten. Im Gegensatz zu diesen reagierten Anwender, die nicht an der Entwicklung teilgenommen hatten, viel positiver auf die neue Software. Somit scheint das CDD ein Ergebnis zu liefern, das zwar nicht jeden einzelnen Anwender vollständig zufriedenstellt, jedoch von der Gesamtnutzerschaft positiv aufgenommen wird. Um in Zukunft auch die einbezogenen Anwender zufriedener zu stimmen, bedarf es einer verbesserten Form der Kommunikationsunterstützung. Diese soll das gegenseitige Verständnis erleichtern, den kreativen Austausch anregen und insbesondere schon

in frühen Entwicklungsstadien eine graphische Repräsentation des Systems ermöglichen¹⁹.

¹⁹ Siehe Kapitel 7.2

8 Fazit

Zu den Voraussetzungen für ein CDD zählen die Verbundenheit der Community sowie die Motivation der beteiligten Anwender und Entwickler, diesen Prozess zu unterstützen. Die Untersuchung zeigt, dass das Gemeinschaftsgefühl innerhalb einer Community vor allem durch das gemeinsame Interesse und den häufigen Kontakt der Mitglieder zustande kommt. Obwohl der Kontakt in einer Nutzer-Community nicht so persönlich und emotional ist wie in manchen anderen Gemeinschaften, fühlen sich gerade die Mitglieder, die sich häufig an Diskussionen beteiligen, miteinander verbunden. Diese Basis ist vorteilhaft, da sowohl Anwender als auch Entwickler, die sich aktiv in einer solchen Community aufhalten, manche Eigenheiten und Kenntnisse der anderen Mitglieder einschätzen können, wodurch sich bei vielen Personen ein Vertrauensverhältnis entwickelt. Gerade dieses Vertrauen ist die Grundlage des CDD-Konzepts. Für eine communitygesteuerte Entwicklung muss das Unternehmen auf die Fähigkeiten seiner Kunden vertrauen. Ebenso müssen die Anwender den Entwicklern glauben, dass sie tatsächlich etwas bewirken können. Dieser Gedanke ist der größte Motivationsfaktor der Anwender. Sie wollen die Möglichkeit nutzen, das Produkt in ihrem Sinne zu verändern. Außerdem motivieren sie der Gemeinschaftsgedanke und der Spaß an der Zusammenarbeit. Die Motivation der Entwickler hingegen begründet sich auf der Möglichkeit, neue Ideen zu erhalten, um ein Produkt entwickeln zu können, das noch stärker am Anwender ausgerichtet ist. Zusätzlich erhoffen sich die Unternehmen eine verstärkte Kundenbindung durch deren Mitwirkung.

Sowohl Anwender als auch Entwickler zeigten sich trotz einiger Schwierigkeiten im Projektverlauf überwiegend zufrieden mit dem Konzept des CDD. Die in der Fallstudie aufgetretenen Probleme wurden zu großen Teilen als unternehmens- oder projektspezifisch eingestuft. Bei einer konsequenten Umsetzung des Modells und Vermeidung solcher Schwierigkeiten sei das CDD ein erfolgversprechender Ansatz. Wichtigste Faktoren dabei seien die Unterstützung der Kommunikation aller Gremien, eine nicht zu eng bemessene Zeitplanung sowie die klare Strukturierung und Organisation des Projektverlaufs. Dann könne vor allem die Weiterentwicklung bestehender Software-Produkte durch innovative Ideen und den vielfältigen

Meinungsaustausch der Community vorangetrieben werden, während gleichzeitig die Bindung der Kunden an das Produkt erhöht werde.

Zur optimalen Unterstützung eines CDD können sehr unterschiedliche Tools eingesetzt werden. Die meisten dienen dabei der Vereinfachung der Kommunikation oder der Anregung zu neuen kreativen Ideen. Verschiedene Tools wurden in den Kapiteln 6.5 und 7.2 vorgestellt. Mittels graphischer Design-Tools kann die Entwicklung bereits von Beginn an unterstützt werden, was Missverständnisse bei abstrakten Ideen verhindert und somit zu mehr Klarheit und weniger Enttäuschungen im Nachhinein führt.

9 Zusammenfassung und Ausblick

9.1 Zusammenfassung

Diese Arbeit befasste sich mit der Einbeziehung von Nutzer-Communities in den Software-Designprozess. Ziel dabei war es, ein neues Konzept zu entwickeln, das es ermöglicht, eine gesamte Nutzer-Community in die Entwicklung von Software-Produkten einzubinden. Dafür wurden zunächst durch eine Analyse der Fachliteratur die grundlegenden Eigenschaften der Bereiche Online-Communities und Software-Entwicklung, darunter speziell Konzepte der Nutzer-Partizipation, erfasst. Anschließend wurden bisherige Ansätze zur Einbeziehung von Nutzer-Communities in Entwicklungsprozesse untersucht, um Probleme und Forschungslücken in diesem Bereich aufzudecken.

Aufbauend auf diesen Ergebnissen wurde der Ansatz des Community Driven Development entwickelt. Durch die Einführung von zwei Vermittlungsgremien, die die Ideen und Meinungen der Community schrittweise bündeln und zu einem Pflichtenheft verdichten, ermöglicht das neue Konzept eine strukturierte Einbeziehung von Nutzer-Communities ins Software-Design. Weitere Schwerpunkte des Ansatzes sind die fortwährende Mitwirkung der Anwender über den gesamten Projektverlauf hinweg sowie eine durch die Nutzer vorangetriebene Entwicklung. Die Anwender genießen somit das Vertrauen des Unternehmens und nehmen eine den Entwicklern gleichgestellte Rolle ein. Durch die Zusammenarbeit zweier Gremien, der Volkskammer und des Zentralrats, können die Ideen und Meinungen der Anwender strukturiert diskutiert und gebündelt werden, um letztlich Entscheidungen im Sinne der Nutzerschaft zu treffen.

Das neue Konzept wurde in einer Fallstudie auf seine Praxistauglichkeit untersucht. Begleitet wurde dazu ein Projekt, beim dem im Rahmen eines CDD die neue Version einer Media Center Software gemeinsam mit einer Nutzer-Community entwickelt wurde. Neben der beobachtenden Begleitung des Projektverlaufs wurden Interviews mit beteiligten Anwendern und Entwicklern durchgeführt, um zuvor aufgestellte Forschungsfragen zu beantworten und Schwierigkeiten im Projekt zu ermitteln.

Aufbauend auf den entdeckten Problemen wurde ein Katalog aus Maßnahmen erstellt, mittels derer die Zusammenarbeit unterstützt und optimiert werden kann. Eine dieser Maßnahmen wurde prototypisch umgesetzt. Dabei handelte es sich um ein in die Beta-Version der Software integriertes Plug-In, über das die Anwender die Möglichkeit hatten, Bewertungen für die Komponenten des Systems abzugeben. Das gesammelte und automatisch in eine Diagrammstruktur umgewandelte Feedback konnte den Entwicklern somit einen Überblick über die Zufriedenheit der Anwender geben und aufzeigen, in welchen Bereichen der Software noch Verbesserungsbedarf herrschte.

Im untersuchten Anwendungsfall stellte vor allem die Kommunikation ein großes Problem dar. Aufgrund unzureichender Planung im Vorfeld des Projekts waren keine klaren Vorgaben zum Projektverlauf bekannt. Auch während der Entwicklung mangelte es an Steuerung, Meilensteine wurden kaum gesetzt und oft nicht eingehalten. Innerhalb des Zentralrats fehlte eine klare Verteilung der Zuständigkeiten. Auch die verwendeten Kommunikationsmedien erwiesen sich als unvorteilhaft. Der Informationsfluss zwischen den beiden Gremien war zu gering, so dass sich viele Anwender aus der eigentlichen Entwicklung ausgeschlossen fühlten. Weitere Schwierigkeiten traten bei zeitlichen Abstimmungen sowie in der Koordination des Projektverlaufs auf. Auch diese Probleme gründeten zu großem Teil auf mangelnder Kommunikation. Ein wesentliches Problem im untersuchten Anwendungsfall war die späte Veröffentlichung des ersten Prototyps der Software. Für eine fortwährende Einbeziehung der Nutzer sollte dies möglichst frühzeitig geschehen. Die im Projekt aufgetretenen Schwierigkeiten wurden zum Großteil als fallspezifisch eingestuft. Um sie in zukünftigen Projekten umgehen zu können, wurden einige organisatorische Verbesserungsmaßnahmen sowie unterschiedliche Tools vorgestellt, die gerade in diesen Bereichen eine gute Unterstützung bieten können.

9.2 Ausblick

Trotz der aufgetretenen Schwierigkeiten erwies sich das eigentliche Konzept des CDD als ein guter Ansatz, der genauer untersucht und weiter optimiert werden sollte. Von nahezu allen Beteiligten wurde das Konzept als positiv und erfolgversprechend eingestuft. Der Großteil sowohl der Anwender als auch der Entwickler würde gerne erneut an entsprechenden Projekten mitwirken.

In zukünftigen Studien sollte genauer untersucht werden, wie erfolgreich ein CDD sein kann, wenn die herausgearbeiteten Verbesserungsmaßnahmen umgesetzt werden. Es bleibt zu prüfen, ob es letztlich für ein Unternehmen rentabel ist, wenn es marktwirtschaftliche Ziele und Fristen zurückstellt, um die Entwicklung tatsächlich durch die Community vorantreiben zu lassen. Um dahingehend Ergebnisse zu erlangen, sollte in folgenden Untersuchungen eine noch größere Zahl an Anwendern im Zentralrat vertreten sein. Diese Nutzer sollten innerhalb des Gremiums zusätzlich wichtige Funktionen, eventuell auch Aufgaben der Projektleitung, übernehmen.

Das Konzept des CDD sollte noch intensiver auf die gestalterische Rolle der Anwender ausgerichtet werden. Zu diesem Zweck könnte der Ansatz durch unterschiedliche Softwaretools unterstützt werden. Es scheint sinnvoll, eine an den Anwendungsfall und die Community angepasste Unterstützungsmöglichkeit zu entwickeln, die kreative Ideenfindung und Diskussion in einem System verbindet. Für die Arbeit der Volkskammer wäre ein graphisches Basteltool denkbar, das den Anwendern die Möglichkeiten bietet, ihre Gedanken zu entfalten und mit Hilfe des Tools visuelle Repräsentationen ihrer Ideen zu entwickeln. Diese sollten automatisch allen Beteiligten zur Ansicht und Diskussion zur Verfügung stehen. Eine einfache Pro/Contra-Bewertung der unterschiedlichen Ideen durch die Anwender könnte die Arbeit des Zentralrats erleichtern, da sich so schnell erkennen ließe, welche Meinung die Mehrheit der Community vertritt und welche Ansätze somit weiter verfolgt werden sollten. Interessante Ideen könnten dann gemeinsam von allen Nutzern zu vollständigen Konzepten weiterentwickelt werden. All diese Schritte sollten dabei von dem Tool unterstützt werden, so dass die Anwender nicht zwischen unterschiedlichen Medien wechseln müssen.

Abkürzungsverzeichnis

AT&T	American Telephone & Telegraph Corporation
CDD	Community Driven Development
CoP	Community of Practice
DATech	Deutsche Akkreditierungsstelle Technik GmbH
FAQ	Frequently Asked Questions
I-DIAG	Interactive Diagenesis
NordiCHI	Nordic Conference on Human-Computer Interaction
PD	Participatory Design
RENISYS	Research Network Information System Specification Method
STEPS	Softwaretechnik für evolutionäre partizipative Systemgestaltung
UCD	User Centered Design
UDD	User Driven Development
VK	Volkskammer
ZR	Zentralrat

Abbildungsverzeichnis

Abbildung 1: Klassifikation von Communities nach Seufert et al. [vgl. SeMo+02]	18
Abbildung 2: Wasserfallmodell nach Royce [vgl. Royc70].....	25
Abbildung 3: Spiralmodell nach Boehm [vgl. Boeh88].....	26
Abbildung 4: Design-Use-Cycle nach Dzida [Geis+06, S. 26]	27
Abbildung 5: Klassifikationsmatrix von Participatory Design Techniken nach Grad der Nutzeraktivität im Design und nach Beginn der Partizipation.....	34
Abbildung 6: CDD-Konzept zur Einbeziehung mehrerer Nutzer-Communities.....	48
Abbildung 7: Zeitstrahl des Projektverlaufs im Anwendungsfall Omega.....	54
Abbildung 8: Beispielforum des Community-Design-Tools Zeno [PiMä+00]	64
Abbildung 9: Client-Server-Architektur der Plug-In-Anbindung.....	76
Abbildung 10: Bewertung des Designs (Stand vom 21.01.2008)	77
Abbildung 11: Bewertung der Funktionalität (Stand vom 21.01.2008).....	77
Abbildung 12: Bewertung der Bedienbarkeit (Stand vom 21.01.2008)	78

Literaturverzeichnis

[AcSw+03] Ackerman, Mark S.; Swenson, Anne; Cotterill, Stephen; DeMaagd, Kurtis: *I-DIAG: From Community Discussion to Knowledge Distillation*. In: *Proceedings of the First Communities and Technologies Conference*, Pages 307-325, Deventer, The Netherlands 2003.

[AlBa+93] Allen, C. Dennis; Ballman, Don; Begg, Vivienne; Miller-Jacobs, Harold H.; Muller, Michael; Nielsen, Jakob; Spool, James: *User Involvement In The Design Process: Why, When & How?* In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Pages 251-254, Amsterdam, The Netherlands 1993.

[AlEd+01] Allen, Jennifer; Edouard, Rodolphe; Gaxiola, George; Verdtzabella, Seyon; Vinh, John: <http://cse.stanford.edu/class/cs201/projects-00-01/participatory-design>, *Participatory Design (PD): An Approach for System Design*, 2001, abgerufen am 13.01.2008.

[ArHa95] Armstrong, Arthur G.; Hagel, John: *The real value of online communities*. In: *Creating value in the network economy. Harvard Business Review Book Series*, Pages 173-185, Boston, Massachusetts, USA 1995.

[Bask99] Baskerville, Richard L.: *Investigating Information Systems with Action Research*. In: *Communications of the AIS*, Volume 2, Issue 3, Article 4, Atlanta, Georgia, USA 1999.

[Boeh88] Boehm, Barry W.: *A spiral model of software development and enhancement*. In: *Computer*, Volume 21, Issue 5, Pages 61-72, Los Alamitos, California, USA 1988.

[BrKu05] Brecht, Rike; Kunert, Tibor: *User Requirements and Design Guidance for Interactive Television News Applications*. In: *Conference Proceedings of EuroITV 2005, 3rd European Conference on Interactive Television*, Aalborg, Denmark 2005.

[Bund04] Bundesministerium des Innern: http://www.kbst.bund.de/chn_047/nn_837872/Content/Standards/V__Modell/vmodell__node.html__nnn=true, *Das neue V-Modell® XT Release 1.2 – Der Entwicklungsstandard für IT-Systeme des Bundes*, 2004, abgerufen am 23.06.2007.

[Chan04] Chandrasekar, Giridhar: <http://www.comp.leeds.ac.uk/mscproj/reports/0304/chandrasekar.pdf>, *Design and Implementation of Easily Maintained Online Community*, 2004, abgerufen am 22.06.2007.

[CoPi+06] Costabile, Maria Francesca; Piccinno, Antonio; Fogli, Daniela; Marcante, Andrea: *Supporting Interaction and Co-evolution of Users and Systems*. In: *Proceedings of the working conference on advances visual interfaces*, Pages 143-150, Venezia, Italy 2006.

[Dzid97] Dzida, Wolfgang: *International User-Interface-Standardization*. In: Tucker, Jr.; Allen, B.: *The Computer Science and Engineering Handbook*, Pages 1474-1493, CRC Press, Boca Raton, Florida, USA 1997.

[Felk05] Christian Felken: <http://www.webmaster-resource.de/tutorials/php/diagramme-mit-php-dynamisch-erstellen-04.php>, *Diagramme mit PHP dynamisch erstellen*, 2005, abgerufen am 03.12.2007.

[Finc01] Finck, Nick: http://www.digital-web.com/articles/derek_powazek/, *Interview mit Derek M. Powazek*, 2001, abgerufen am 11.03.2007.

[FIRe+89] Floyd, Christiane; Reisin, F.-M.; Schmidt, G.: *STEPS to Software Development with Users*. In: Ghezzi, C., McDermid J.A.: *ESEC '89, Lecture Notes in Computer Science*, No. 387, Pages 48-64, Springer Verlag, Berlin/Heidelberg, Germany 1989.

[Ford94] Ford, Shannon: *There's No Place Like Home, The Doors of Perception 2 Conference*. In: *ACM SIGCHI Bulletin*, Volume 27, Issue 3, Pages 26-29, Amsterdam, The Netherlands 1994.

[FrHi+05] Franke, Nikolaus; von Hippel, Eric; Schreier, Martin: *Finding commercially attractive user innovations: A test of lead user theory*. In: *Journal of Product Innovation Management*, MIT Sloan School of Management, Massachusetts, USA 2005.

[FüBa+06] Füller, Johann; Bartl, Michael; Ernst, Holger; Mühlbacher, Hans: *Community based innovation: How to integrate members of virtual communities into new product development*. In: *Electronic Commerce Research*, Volume 6, Issue 1, Pages 57-73, Norwell, Massachusetts, USA 2006.

[GaDu+99] Gaver, Bill; Dunne, Tony; Pacenti, Elena: *Design: Cultural Probes*. In: *ACM Interactions*, Volume 6, Issue 1, Pages 21-29, New York, USA 1999.

[Geis+06] Geis, Thomas; Deutsche Akkreditierungsstelle Technik GmbH, Arbeitsgruppe Usability Engineering und Software-Ergonomie: http://www.datech.de/share/files/Pruefhandbuch_ISO_13407.pdf, *DATEch-Prüfhandbuch Usability-Engineering-Prozess, Leitfaden für die Evaluierung des Usability-Engineering-Prozesses bei der Herstellung und Pflege von interaktiven Systemen auf der Grundlage von DIN EN ISO 13407*, Version 1.4, 2006 – abgerufen am 15.05.2007.

[HaAr97] Hagel, John; Armstrong, Arthur G.: *Net Gain – Expanding Markets Through Virtual Communities*. Harvard Business School Press, Boston, Massachusetts, USA 1997.

[Hans02] Hansson, Christina: [http://www.bth.se/fou/cuppsats.nsf/all/ecd29a89d91c8674c1256c3d0036226d/\\$file/UserDrivenFinalVersion.pdf](http://www.bth.se/fou/cuppsats.nsf/all/ecd29a89d91c8674c1256c3d0036226d/$file/UserDrivenFinalVersion.pdf), *User Driven Software Development in a Small Company*, Master-Thesis, Tekniska Högskola, Blekinge, Sweden 2002, abgerufen am 09.04.2007.

- [Hipp86] von Hippel, Eric: *Lead Users: A Source of novel product concepts*. In: *Management Science*, Volume 32, Pages 791-805, Cambridge, Massachusetts, USA 1986.
- [Hipp02] von Hippel, Eric: *User Toolkits for Innovation: A Practical Guide*. MIT Sloan School of Management Working Paper, Cambridge, Massachusetts, USA 2002.
- [Hippel] von Hippel, Eric: <http://web.mit.edu/evhippel/www/index.html>, Eric von Hippels Homepage, Cambridge, Massachusetts, USA.
- [Hoon07] Hoonhout, Jettie: *ExperienceLabs: investigating people's experiences in realistic "lab" settings*. In: *Proceedings of the 2007 conference on designing pleasurable products and interfaces*, Pages 511-515, ACM, New York, NY, USA 2007.
- [HuLe80] Hult, Margareta; Lennung, Sven-Ake: *Towards A Definition of Action Research: A Note and Bibliography*. In: *Journal of Management Studies*, Volume 17, Issue 2, Pages 241-250, 1980.
- [HuMa+04] Hulkko, Sami; Mattelmäki, Tuuli; Virtanen, Katja; Keinonen, Turkka: *Mobile Probes*. In: *ACM International Conference Proceedings Series, Volume 82, Proceedings of the third Nordic Conference on Human-Computer Interaction*, Pages 43-51, Tampere, Finland 2004.
- [IrEr+04] Irestig, Magnus; Eriksson, Henrik; Timpka, Toomas: *The Impact of Participation in Information System Design: A Comparison of Contextual Placements*. In: *Proceedings of the eighth Conference on Participatory Design: Artful Integration: interweaving media, materials and practices*, Volume 1, Pages 102-111, Toronto, Ontario, Canada 2004.
- [JäMa03] Jääskö, Vesa; Mattelmäki, Tuuli: *Observing and Probing*. In: *Proceedings of the 2003 International Conference on Designing Pleasurable Products and Interfaces*, Pages 126-131, Pittsburgh, Pennsylvania, USA 2003.
- [Jann06] Janneck, Michael: *Gebrauchstaugliche didaktische Software – Entwicklungsprozess, Didaktik, Gestaltungsprinzipien*. Dissertation, Universität Hamburg, Department Informatik, Hamburg, Germany 2006.
- [JoGr06] Jonsdatter, Gitte; Gregory, Judith: *Innovation Play: Visualization and Storytelling for Engaging Participation in Design*. In: *Participatory Design Conference 2006*, noch keine Veröffentlichung erschienen, Trento, Italy 2006.
- [KaCh+00] Kannan, P. K.; Chang, Ai-Mei; Whinston, Andrew B.: *Electronic Communities in E-Business: Their Role and Issues*. In: *Information Systems Frontiers*, Volume 1, Issue 4, Pages 415-426, Hingham, Massachusetts, USA 2000.
- [KeBl98] Kensing, Finn; Blomberg, Jeanette: *Participatory Design: Issues and Concerns*. In: *Computer Supported Cooperative Work*, Volume 7, Issue 3-4, Pages 167-185, Kluwer Academic Publishers, Norwell, Massachusetts, USA 1998.

[LaPr02] Lazar, Jonathan; Preece, Jennifer: *Social Considerations in Online Communities: Usability, Sociability, and Success Factors*. In: van Oostendorp, Herre: *Cognition in a Digital World*. Lawrence Erlbaum Associates Inc. Publishers, Mahwah, New Jersey, USA 2002.

[Lech07] Lechner, Ulrike: [http://wi.informatik.unibw-muenchen.de/C15/lectures-digitaleMed\(HT2007\)/Document%20Library/Digitale%20Medien__Teil%208__071122__V1__ule.pdf](http://wi.informatik.unibw-muenchen.de/C15/lectures-digitaleMed(HT2007)/Document%20Library/Digitale%20Medien__Teil%208__071122__V1__ule.pdf), *Open Innovation*, zu [RePi06], Germany 2007, abgerufen am 29.01.2008.

[LeMa04] Letondal, Catherine; Mackay, Wendy E.: *Participatory Programming and the Scope of Mutual Responsibility: Balancing Scientific, Design and Software Commitment*. In: *Proceedings of the eighth Conference on Participatory Design: Artful Integration: interweaving media, materials and practices – Volume 1*, Pages 31-41, Toronto, Ontario, Canada 2004.

[Levi98] Levinger, David: <http://trout.cpsr.org/conferences/pdc98/history.html>, *Participatory Design History*, San Francisco, USA 1998, abgerufen am 13.01.2008.

[MaPe+04] Macdonald, Nico; Perks, Martyn; Reimann, Robert; Oppenheimer, Aaron: *Beyond Human-Centered Design? In: Symposium on Designing Interactive Systems: processes, practices, methods and techniques*, Pages 373-374, Cambridge, Massachusetts, USA 2004.

[Merz02] Merz, Helmut: <http://www.cyber-communities.de/?pg=artikel>, *Cyber-Communities – virtuelle Gemeinschaften im Internet*, 2002, abgerufen am 18.04.2007.

[MiMä05] Mittilä, Tuula; Mäntymäki, Maria: *The End of Attraction: Why People Stop Visiting Company Online Communities?* In: *Proceedings of eBRF 2005 Conference*, Pages 25-37, Tampere, Finland 2005.

[MoRi96] de Moor, Aldo; van der Rijst, Nardo B. J.: *Fostering Active User Involvement in the Specification of Network Information Systems*. In: *Proceedings of the 1996 Interdisciplinary Information Science Conference*, Pages 105-118, Delft, The Netherlands 1996.

[MuWi+92] Muller, Michael J.; Wildman, Daniel M.; White, Ellen A.: *Taxonomy of Participatory Design Practices*. In: *Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems*, Page 34, Monterey, California, USA 1992.

[Omega-Forum] – Quelle durch Anonymisierung nicht angegeben.

[Orma05] Orman, Levent V.: *Design and Implementation of Electronic Communities*. In: *Information Technology and Management*, Volume 6, Issue 2-3; Pages 163-179, Hingham, Massachusetts, USA 2005.

[PaSi94] Pagel, Bernd-Uwe; Six, Hans-Werner: *Software Engineering – Die Phasen der Software-Entwicklung*. Band 1, Addison-Wesley Publishing Company, Bonn, Germany 1994.

[Pelk00] Pelka, Anita Isabella: http://v-modell.iabg.de/index.php?option=com_docman&task=doc_view&gid=35, *Integration von Software- und Usability-Engineering am Beispiel des V-Modells*. Diplomarbeit, Friedrich-Wilhelms-Universität Bonn, Institut für Informatik, Bonn, Germany 2000, abgerufen am 11.07.2007.

[PeRo84] Peters, M.; Robinson, V.: *The Origins and Status of Action Research*. In: *Journal of Applied Behavioral Science*, Volume 20, Issue 2, Pages 113-124, 1984.

[PiMä+00] Pipek, Volkmar; Märker, Oliver; Rinner, Claus; Schmidt-Belz, Barbara: *Discussions and Decisions: Enabling Participation in Design in Geographical Communities*. In: Gurstein, M. (ed.): *Community Informatics: Enabling Communities with Information and Communications Technologies*, IDEA Publishing Group, Pages 358-375, Germany 2000.

[Ploe01] Ploegert, Klaus: <http://www.softwaretesting.de/article/view/36/1/7/>, *Das V-Modell*, 2001, abgerufen am 26.06.2007.

[Powa01] Powazek, Derek M.: *Design for Community – The Art of Connecting Real People in Virtual Places*. New Riders, USA 2001.

[Pree00] Preece, Jennifer: *Online Communities. Supporting Sociability, Designing Usability*. John Wiley and Sons Ltd, England 2000.

[PrRo+02] Preece, Jennifer; Rogers, Y.; Sharp, H.: *Interaction Design: Beyond Human-Computer Interaction*. John Wiley and Sons Ltd, England 2002.

[RaMo+93] Rauterberg, Matthias; Mollenhauer, Raimund; Spinus, Philipp: *Phasenmodell ist OUT – Benutzerbeteiligung jetzt auch bei Standardsoftware-Entwicklung*. In: *Berichte des German Chapter of the ACM*, Volume 40, Pages 572-591, Teubner Verlag, Germany 1993.

[RePi06] Reichwald, Ralf; Piller, Frank: *Interaktive Wertschöpfung*. Gabler Verlag, Germany 2006.

[Rhei93] Rheingold, Howard: *The Virtual Community – Homesteading on the Electronic Frontier*. Addison-Wesley, Reading, Massachusetts, USA 1993.

[Royc70] Royce, Winston W.: *Managing the Development of Large Software Systems*. In: *IEEE WESCON*, S. 1-9, Los Angeles, California, USA 1970. - Nachgedruckt in: *Proceedings of the 9th International Conference on Software Engineering*, Pages 328-338, Monterey, California, USA 1987.

[SeMo+02] Seufert, Sabine; Moisseeva, Marina; Steinbeck, Reinhold: *Virtuelle Communities gestalten*. In: Hohenstein, A.; Wilbers, K.: *Handbuch E-Learning*. Fachverlag Deutscher Wirtschaftsdienst, Köln, Germany 2002.

[Shne98] Shneiderman, Ben: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 3rd Edition, Addison-Wesley, Reading, Massachusetts, USA 1998.

[ThWi92] Thompson, Michael; Wishbow, Nina: *Prototyping: tools and techniques: improving software and documentation quality through rapid prototyping*. In: *Proceedings of the 10th annual international conference on systems documentation*, Pages 191-1999, Ottawa, Ontario, Canada 1992.

[TsSt+97] Tsai, Bor-Yuan; Stobart, Simon; Parrington, Norman; Thompson, Barrie: *Iterative design and testing within the software development life cycle*. In: *Software Quality Control*, Volume 6, Issue 4, Pages 295-310, Kluwer Academic Publishers, Hingham, Massachusetts, USA 1997.

[WaAt+06] Wania, Christine E.; Atwood, Michael E.; McCain, Katherine W.: *How do Design and Evaluation Interrelate in HCI Research?* In: *Proceedings of the 6th ACM Conference on Designing Interactive Systems*, Pages 90-98, University Park, PA, USA 2006.

[Well98] Wellman, Barry: *Virtual Community: Introducing a New SIGGROUP Focus Area*. In: *ACM SIGGROUP Bulletin*, Volume 19, Issue 1, Pages 18-20, New York, USA 1998.

[Wied06] Wiedenhöfer, Torben: *Help in Context – Konzeption und Umsetzung eines community-unterstützenden Hilfesystems*. Diplomarbeit, Universität Siegen, Germany 2006.

[YuJi+07] Yu, Jie; Jiang, Zhenhui; Chan, Hock Chuan: *Knowledge contribution in problem solving virtual communities: the mediating role of individual motivations*. In: *Proceedings of the 2007 ACM SIGMIS CPR Conference on 2007 computer personnel doctoral consortium and research conference: The global information technology workforce*, Pages 144-152, New York, USA 2007.

Anhang

Der Anhang der Diplomarbeit befindet sich auf der beiliegenden CD.

Diplomarbeit.pdf

Digitale Version der Diplomarbeit

Interviews.pdf

Fragebögen für die Interviews mit Anwendern und Entwicklern

Protokolle der Interviews mit Anwendern und Entwicklern

Mitgliederstatistik.pdf

Statistik über die Zahl und Aktivität der Forumsmitglieder

Omega Bulle.pdf

Veröffentlichung des Unternehmens zu den Zielen und grundlegenden Festlegungen für die der Entwicklung von OmegaTV4

Statistik offene Diskussion.pdf

Statistik über die offene Diskussion zu OmegaTV4

Statistik Volkskammer-Diskussion.pdf

Statistik über die Diskussion im Unterforum der Volkskammer

Plug-In [Ordner]

Implementierung und Ergebnisse des im Rahmen dieser Arbeit erstellten Plug-Ins

Der Ordner enthält eine Datei **Inhalt.pdf**, die eine Übersicht und genauere Erläuterungen zu den übrigen Inhalten des Ordners bietet.

Erklärung

Eidesstattliche Erklärung

Ich erkläre hiermit eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und alle Abschnitte, die wörtlich oder annähernd wörtlich aus einer Veröffentlichung entnommen sind, als solche kenntlich gemacht habe, ferner, dass die Arbeit noch nicht veröffentlicht und auch keiner anderen Prüfungsbehörde vorgelegt worden ist.

Siegen, den 21. Februar 2008

Sinja Offenberg