

Michael Richter Markus Flückiger

Usability Engineering kompakt

Benutzbare Software gezielt entwickeln

2. Auflage



Spektrum
AKADEMISCHER VERLAG

Usability Engineering kompakt

Werke der „kompakt-Reihe“ zu wichtigen Konzepten und Technologien der IT-Branche:

- ermöglichen einen raschen Einstieg,
- bieten einen fundierten Überblick,
- sind praxisorientiert, aktuell und immer ihren Preis wert.

Bisher erschienen:

- Heide Balzert
UML kompakt, 2. Auflage
- Andreas Böhm / Elisabeth Felt
e-commerce kompakt
- Christian Bunse / Antje von Knethen
Vorgehensmodelle kompakt, 2. Auflage
- Holger Dörnemann / René Meyer
Anforderungsmanagement kompakt
- Christof Ebert
Outsourcing kompakt
- Christof Ebert
Risikomanagement kompakt
- Karl Eilebrecht / Gernot Starke
Patterns kompakt, 3. Auflage
- Andreas Essigkrug / Thomas Mey
Rational Unified Process kompakt, 2. Auflage
- Peter Hruschka / Chris Rupp / Gernot Starke
Agility kompakt, 2. Auflage
- Arne Koschel / Stefan Fischer / Gerhard Wagner
J2EE/Java EE kompakt, 2. Auflage
- Michael Kuschke / Ludger Wölfel
Web Services kompakt
- Torsten Langner
C# kompakt
- Pascal Mangold
IT-Projektmanagement kompakt, 3. Auflage
- Michael Richter / Markus Flückiger
Usability Engineering kompakt, 2. Auflage
- Thilo Rottach / Sascha Groß
XML kompakt: die wichtigsten Standards
- SOPHIST GROUP / Chris Rupp
Systemanalyse kompakt, 2. Auflage
- Gernot Starke / Peter Hruschka
Software-Architektur kompakt
- Ernst Tiemeyer
IT-Controlling kompakt
- Ernst Tiemeyer
IT-Servicemanagement kompakt
- Ralf Westphal
.NET kompakt
- Ralf Westphal / Christian Weyer
.NET 3.0 kompakt

Michael Richter / Markus Flückiger

Usability Engineering kompakt

Benutzbare Software gezielt entwickeln

2. Auflage

Spektrum
AKADEMISCHER VERLAG

The logo for Spektrum Akademischer Verlag features the word "Spektrum" in a large, bold, serif font. Below it, the words "AKADEMISCHER VERLAG" are written in a smaller, all-caps, sans-serif font. A horizontal grey bar is positioned directly beneath the text.

Autoren:

Michael Richter und Markus D. Flückiger
Zühlke Engineering AG, 8952 Schlieren, Schweiz
E-Mail: Michael Richter: rim@zuehlke.com
Markus Flückiger: mdf@zuehlke.com

Wichtiger Hinweis für den Benutzer

Der Verlag und die Autoren haben alle Sorgfalt walten lassen, um vollständige und akkurate Informationen in diesem Buch zu publizieren. Der Verlag übernimmt weder Garantie noch die juristische Verantwortung oder irgendeine Haftung für die Nutzung dieser Informationen, für deren Wirtschaftlichkeit oder fehlerfreie Funktion für einen bestimmten Zweck. Ferner kann der Verlag für Schäden, die auf einer Fehlfunktion von Programmen oder ähnliches zurückzuführen sind, nicht haftbar gemacht werden. Auch nicht für die Verletzung von Patent- und anderen Rechten Dritter, die daraus resultieren. Eine telefonische oder schriftliche Beratung durch den Verlag über den Einsatz der Programme ist nicht möglich. Der Verlag übernimmt keine Gewähr dafür, dass die beschriebenen Verfahren, Programme usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Der Verlag hat sich bemüht, sämtliche Rechteinhaber von Abbildungen zu ermitteln. Sollte dem Verlag gegenüber dennoch der Nachweis der Rechtsinhaberschaft geführt werden, wird das branchenübliche Honorar gezahlt.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer ist ein Unternehmen von Springer Science+Business Media
springer.de

2. Auflage 2010

© Spektrum Akademischer Verlag Heidelberg 2010

Spektrum Akademischer Verlag ist ein Imprint von Springer

10 11 12 13 14 5 4 3 2 1

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Planung und Lektorat: Dr. Andreas Rüdinger, Barbara Lühker

Herstellung und Satz: Crest Premedia Solutions (P) Ltd, Pune, Maharashtra, India

Umschlaggestaltung: SpieszDesign, Neu-Ulm

ISBN 978-3-8274-2328-3

Vorwort

Zur zweiten Auflage

Beispiele spielen eine wichtige Rolle in unserer täglichen Kommunikation. Sie unterstützen das Lernen von Sachverhalten und schaffen ein gemeinsames Verständnis. Beispiele sind auch nützliche Werkzeuge in Software- oder Produktentwicklungsprojekten. Mit aussagekräftigen Beispielen können Informationen auf äußerst effiziente Weise von der Analyse bis zur Implementierung transportiert werden. Usability-Methoden nutzen diesen Sachverhalt. Sie bringen die Bedürfnisse der Benutzer in die Entwicklung ein, erlauben Feedback zu Lösungsvarianten und fokussieren auf die Realität der täglichen Anwendung von Produkten und Systemen, indem sie konkrete Beispiele verwenden.

In der zweiten Auflage dieses Buches möchten wir Ihnen ebenfalls mehr Beispiele mitgeben. Ein ganzes Kapitel ist vier interessanten Fallstudien aus der Praxis gewidmet, in denen wir benutzerorientierte Vorgehensweisen und die Anwendung von Usability-Methoden in unterschiedlichen Projekten beleuchten. Zudem haben wir viele Abschnitte überarbeitet, aktualisiert und nach neusten Erkenntnissen erweitert.

Der Erfolg von *Usability Engineering kompakt* freut uns und zeigt, dass das Thema auch im deutschen Sprachraum auf großes Interesse stößt. Wir bedanken uns an dieser Stelle herzlich bei Allen, die zu der vorliegenden zweiten Auflage beigetragen haben, insbesondere Christian, Hilmar, Jeannine, Médard, Olivier, Thomas und Thorsten.

Zur ersten Auflage

In unserer Tätigkeit als Berater werden wir immer wieder nach einer „kurzen Übersicht“ über das Thema Usability Engineering gefragt. Es bleibt uns jeweils nichts anderes übrig, als auf eine Reihe guter Fachbücher zu verweisen. Diese sind meist auf Englisch geschrieben und häufig mehrere hundert Seiten dick. Zudem haben wir den Eindruck, dass viele dieser Bücher für Usability-Fachpersonen selbst geschrieben wurden und Methoden oder Prozesse im Detail behandeln, ohne eine Integration in bestehende Software-Engineering-Vorgehensmodelle zu versuchen.

Dieses Buch richtet sich in erster Linie an Beteiligte im Software-Entwicklungsprozess – Produktverantwortliche, Projektleiter, Berater und Analysten, die vor einer großen Herausforderung stehen: Software oder Produkte zu entwickeln, die gut benutzbar sind. Unser Ziel ist es, Ihnen eine kompetente Übersicht zu verschaffen.

Was erwartet Sie in dieser Lektüre? Lassen Sie uns damit beginnen, was Sie in diesem Buch *nicht* finden werden:

- Regeln, deren Einhaltung gute Usability garantiert
- Eine Anleitung zur Durchführung von Usability-Methoden
- User-Interface-Richtlinien für Entwickler
- Theorie, die in der Praxis nicht anwendbar ist

Sie werden hingegen Antworten zu folgenden Fragen finden:

- Wie viel Usability ist notwendig?
- Wie passen Usability-Aktivitäten in den Entwicklungsprozess?
- Wie plane ich Usability-Methoden ein und wie laufen sie ab?
- Welche Resultate erhalte ich und wie kann ich diese kontrollieren?
- Wie kann ich Benutzerorientierung im Unternehmen etablieren?
- Welche verwandten Gebiete gibt es und wo finde ich weitere Informationen?

Wir hoffen, auch dem interessierten Laien eine leicht verständliche Einführung in die Materie zu vermitteln. Als Benutzer von technischen Systemen haben wir alle die Wahl: Entweder wir akzeptieren, was wir täglich vorgesetzt bekommen, oder wir tun dies nicht. Natürlich können wir nicht in jedem Fall ein anderes System verwenden, wenn uns dies notwendig erscheint. Aber dann können wir zumindest versuchen, zu einer Verbesserung beizutragen.

Unser Dank gilt der Firma Zühlke Engineering, die uns das Schreiben dieses Buches ermöglichte, und allen Personen, die uns dabei mit viel Zeit und Geduld geholfen haben: Andreas, Arun, Barbara, Calla, Christian, Christian, Elisabeth, Jörg, Luana, Marco, Michael, Niko, Patrick, Peter, Rainer, Sandra und Toni.

Wir wünschen Ihnen viel Spaß beim Lesen!

Markus Flückiger und Michael Richter

Inhalt

Vorwort	V
Einführung	1
Wir alle sind Benutzer	1
Der Benutzer ist nicht wie ich	2
Was ist Usability Engineering?	3
Ein Blick in die Vergangenheit	8
Usability Engineering im Entwicklungsprozess	10
Software Engineering: Die vergessenen Benutzer?	10
Usability-Methoden im Zusammenhang	14
Die 7±2 wichtigsten Usability-Methoden	21
Facetten der Arbeit: Contextual Inquiry	22
Modellierte Realität: Personas und Szenarien	28
Einfach kommunizieren: Storyboards	36
Kritzeln für Fortgeschrittene: UI Prototyping	40
Für die Entwicklung dokumentieren: Use Cases	48
Usability Guidelines und Styleguides	54
Auf dem Prüfstand: Usability Testing	60
Zahlenmaterial: Fragebögen	67
Usability im Griff: Planung	77
Ziele erreichen	77
Risiken kontrollieren	78
Rahmenbedingungen	79
Planungsbeispiele	80
Einsatz von Benutzern	84
Strategische Usability	86
Usability unternehmensweit	86
Aufbau eines benutzerorientierten Prozesses	87
Usability-Standardisierung	89
Usability-Institutionalisierung	92
Wie sieht es in Ihrem Unternehmen aus?	93
That's life: Beispiele aus der Praxis	96
Fallstudien	96
Schwierige Situationen	111

Rückblick	117
Ausblick	121
Accessibility	121
User Experience	122
Interaction Design	122
Security und Usability	124
Web Usability	124
Der allgegenwärtige Computer	125
Glossar	128
Literatur	132
Index	137

Einführung

Der Weltraum – unendliche Weiten. Wir schreiben das Jahr 2200. Dies sind die Abenteuer des Raumschiffs Enterprise. Persönliches Logbuch des Captains: Der Computer versteht einfach nicht, was ihm gesagt wird. Scotty versucht seit Tagen, die Bedienung des neuen Transporters in den Griff zu bekommen, und unser Tricorder liefert immer dieselbe unverständliche Fehlermeldung...

Wir alle sind Benutzer

Ist Ihnen auch schon aufgefallen, dass im Fernsehen die Leute immer mühelos mit der Technik klarkommen? Wir hingegen stolpern bei Anwendungsprogrammen, tippen falsche PIN-Codes ein, verlaufen uns in Flughäfen und verzweifeln an neuen Geräten. Im täglichen Kontakt mit technischen Systemen haben wir uns alle schon eine Vorstellung davon gemacht, was Usability bedeutet. Lassen Sie uns diese Einführung deshalb mit einigen Usability-Klassikern aus dem Alltag der Gegenwart beginnen. Sicher sind auch Ihnen schon solche oder ähnliche Situationen mit gut oder schlecht benutzbaren technischen Systemen in Erinnerung geblieben:

- Der Fahrkartenautomat, der immer gut funktionierte, bis zu dem Zeitpunkt, als Sie die Fahrkarte (mit Quittung) für den nächsten Tag lösen wollten.
- Der neue Harddisk-Videorekorder, der auf Knopfdruck das Fußballspiel aufzeichnet, wenn der Pizza-Kurier klingelt. Oder war es eine Tastenkombination? Und wo war noch mal die Anleitung?
- Die Leichtigkeit, mit der Sie Musik aus dem Internet herunterladen, in Musiklisten ordnen und auf Ihren MP3-Player übertragen.
- Die Telefonrechnung, nachdem Sie mit dem neuen automatischen Buchungssystem endlich Ihre Kinotickets für die Abendvorstellung reserviert hatten.

Interaktive Produkte begleiten uns in unserem Alltag. Wahrscheinlich gehören auch Sie zu jenen Menschen, die sich längst damit abgefunden haben, dass viele Systeme schlichtweg kaum zu benutzen, andere dagegen hervorragend sind. Ist das Zufall? Welche Faktoren bestimmen, ob wir mit einem Produkt sehr einfach, nur schwer oder gar nicht zum Ziel kommen? Welche Möglichkeiten bieten sich, diese Faktoren in der Entwicklung von Software und Produkten systematisch in

den Griff zu bekommen? Mit diesen Fragestellungen beschäftigt sich Usability Engineering.

Der Benutzer ist nicht wie ich

Bestimmt haben Sie schon einmal einen wichtigen Text geschrieben und den Entwurf jemand anderem zum Lesen gegeben. Sicher haben Sie die Erfahrung gemacht, wie wertvoll die Hinweise dieser anderen Person waren. Sie selbst hatten sich über längere Zeit intensiv mit dem Thema befasst und waren deshalb nicht mehr in der Lage, sich in die Sicht eines außenstehenden Lesers zu versetzen. Sie hätten den Text auch einfach alleine schreiben können, er wäre allerdings nicht so gut geworden wie nach Einarbeitung des Feedbacks.

Die Entwicklung von Software oder interaktiven Produkten ist (in aller Regel) komplexer als das Verfassen eines Textes. Die Projektbeteiligten sind vom Blickwinkel der späteren Anwender in zweierlei Hinsicht weit entfernt:

- Sie sind Spezialisten, die sich über längere Zeit mit der eingesetzten Technologie befasst haben und die Sichtweise eines unbedarften Benutzers nicht mehr ohne Weiteres einnehmen können.
- Sie sind bezüglich des Anwendungsgebietes, in dem die entwickelte Lösung zum Einsatz kommt, oft Laien. Hier ist der Benutzer der Experte. Die Entwickler werden sich nicht umfänglich mit dem Fachgebiet, den Konzepten und Begriffen und schon gar nicht mit den konkreten Abläufen in der alltäglichen Anwendung vertraut machen können.

In beiden Punkten ist die Perspektive der Benutzer notwendig, damit eine brauchbare Lösung entstehen kann. Usability Engineering befasst sich im Wesentlichen damit, wie die Benutzersicht systematisch in die Entwicklung einbezogen werden kann.

Hintergrund: Perspektivenübernahme

Als *Perspektivenübernahme* wird in der Psychologie die Fähigkeit bezeichnet, eine bestimmte Gegebenheit aus der Sicht eines anderen zu verstehen. Die Fähigkeit der Perspektivenübernahme entwickelt sich im Kindesalter und wird im Verlauf des Lebens individuell unterschiedlich stark ausgeprägt. Dabei spielt es nicht nur eine Rolle, ob man sich in die Lage eines anderen versetzen kann. Entscheidend ist auch, den Bedarf für eine Perspektivenübernahme zu erkennen, die Lage aus der Sicht des anderen zu analysieren und die daraus resultierenden Erkenntnisse anzuwenden.

Was ist Usability Engineering?

Alles sollte so einfach wie möglich gemacht werden, aber nicht einfacher.

(Albert Einstein)

Der Begriff Usability

Noch vor zwei Jahrzehnten war *Usability* als Begriff, zumindest im deutschen Sprachraum, weitgehend unbekannt. Eine eher kleine Fachgemeinde von „Software-Ergonomen“ befasste sich mit der „Gebrauchstauglichkeit“ bei der Verwendung von interaktiven Systemen. Für unvoreingenommene Ohren müssen diese Bezeichnungen klingen, als wären sie dem Labor einer Forschungseinrichtung entwichen. Auch der etwas geläufigere Ausdruck „Benutzerfreundlichkeit“ ist nicht ganz befriedigend, da der Grad der Freundlichkeit eines Systems gegenüber dem Benutzer ein eher diffuses Konzept darstellt. Die „Benutzbarkeit“ eines Systems ist präziser und scheint uns im Sinne der Allgemeinverständlichkeit die treffendste Übersetzung. Usability hat sich inzwischen aber auch im deutschen Sprachraum als Begriff in der alltäglichen Konversation, in den Medien oder auf Produktbeschreibungen etabliert. Wir verwenden die genannten Bezeichnungen in diesem Buch gleichbedeutend.

Usability ist mehr als die Qualität der Benutzeroberfläche

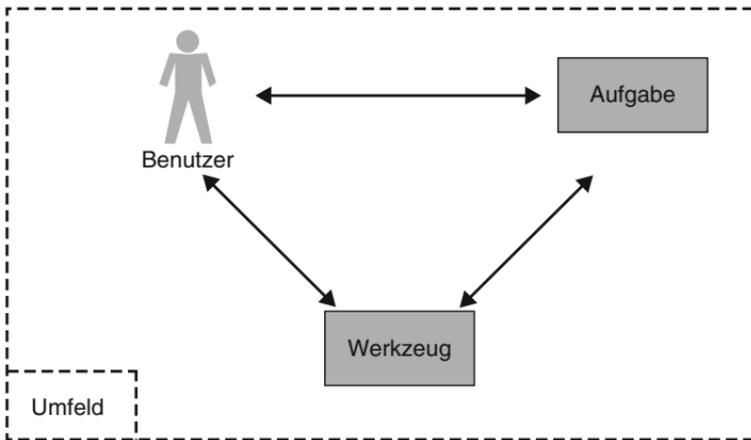
Es gibt viele Definitionen, was Usability ist, und wir wollen an dieser Stelle auch keine weiteren formal korrekten und allgemeingültigen Konzepte postulieren. Für die Einbettung dieses Buches ist eine Begriffsbestimmung indessen notwendig.

Usability wird manchmal im engeren Sinne als Gütekriterium für die Gestaltung einer Benutzeroberfläche verstanden. Qualitätskriterien sind etwa die Anordnung von Bedienelementen, die Anzahl notwendiger Klicks oder die Verständlichkeit der angezeigten Bezeichnungen und Dialoge.

Hinter dem Begriff verbirgt sich jedoch mehr. Die Benutzbarkeit eines Systems muss im Kontext seiner Verwendung beurteilt werden. Software-Anwendungen oder Produkte weisen eine hohe Usability auf, wenn sie von den vorgesehenen Benutzern einfach erlernt und effizient verwendet werden können und diese damit ihre beabsichtigten Ziele und Aufgaben zufriedenstellend ausführen können.

Ein gutes Beispiel, um den Unterschied zwischen engerem und weiterem Verständnis von Usability zu verdeutlichen, ist das Versenden von Kurznachrichten mit Mobiltelefonen. Niemand wird bestreiten, dass rein numerische Tastaturen eigentlich nicht für die Erfassung von Text vorgesehen sind. Viele Benutzer empfinden die Erstellung von Nachrichten damit sogar als ziemlich umständlich. Die Benutzerschnittstelle im engeren Sinne ist für sie nicht optimal. Betrachtet man die gesamte Anwendung, bietet sie hingegen genau das, was der Benutzer eigentlich will: Kurze Nachrichten können auf einfache und effiziente Weise mit mobilen Geräten übermittelt werden. Das Ziel des Benutzers wird vom System gut unterstützt. Mit anderen Worten, die Anwendung insgesamt weist eine gute Usability auf.

Die folgende Abbildung zeigt die vier prinzipiellen Komponenten eines Mensch-Maschine-Systems:



Eine Definition von Usability in diesem weiteren Sinne wurde in einer ISO-Norm festgelegt. Diese Definition wird oft zitiert, und Sie sollten sie deshalb kennen. Die ISO-Norm 9241-11 definiert *Gebrauchstauglichkeit* als „das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“ [DIN EN ISO 9241-11, 1998].

Aus dieser Definition lässt sich ableiten, dass die verbreitete Ansicht, Usability sei ausschließlich eine Eigenschaft eines Produktes,

falsch ist. Um das an einem sehr einfachen Beispiel zu verdeutlichen: Die Usability eines Hammers zum Einschlagen von Nägeln kann gut sein. Doch sie wird ziemlich schlecht ausfallen, wenn Ihre Aufgabe darin besteht, Schrauben einzudrehen. Usability steht dafür, wie gut Benutzer ein Werkzeug in ihrem Umfeld zur Bewältigung ihrer Aufgaben einsetzen können. Entsprechend muss das zu erstellende Produkt in die Welt der Benutzer eingepasst werden.

Die folgende Abbildung veranschaulicht, dass Usability auf verschiedenen Ebenen optimiert werden kann:

Für die Erstellung einer benutzbaren Lösung heißt das, dass die Optimierung der Benutzeroberfläche allein nicht ausreicht. Folgende Aufgaben sind essenziell:

- Analyse der Benutzer, ihrer Aufgaben und des Anwendungskontexts
- Definition des Funktionsumfangs und der benötigten Informationen
- Erarbeitung der optimalen Abläufe und Prozesse



In diesem Zusammenhang werden wir in diesem Buch öfter auf *Abläufe aus Benutzersicht* zu sprechen kommen, um den zeitlichen Aspekt von Interaktionen zu verdeutlichen. Stellen Sie sich die Benutzung eines Systems vereinfacht als Schritte in einer bestimmten, wiederkehrenden Abfolge vor, die zum Ziel führen. Wahrscheinlich gibt es nicht nur eine, sondern mehrere solche Abfolgen. Aus Benutzersicht können diese Abläufe mehr oder weniger gut mit dem System ausgeführt werden. Usability umfasst die Eignung eines Systems, die Abläufe aus Benutzersicht zu unterstützen.

Usability Engineering: Reduktion auf das Wesentliche

Usability Engineering umfasst Mittel und Techniken, um bei der Entwicklung neuer Software oder Produkte die angestrebte Usability zu erreichen. Dies beinhaltet die Fragestellung, wer die genaue Benutzergruppe ist, die Analyse der Arbeitsabläufe sowie die Festlegung der idealen Funktionalität und Konzeption einer passenden Benutzerschnittstelle.

Eine wesentliche Aufgabe von Usability Engineering ist es, unnötige Komplexität zu vermeiden und die Funktionalität eines Produktes auf ein für den Benutzer ideales Minimum zu reduzieren. Das technische System soll den Anwender in der Ausführung seiner Ziele optimal unterstützen und wird genau dafür konzipiert. Diese Reduktion auf das Wesentliche kommt nicht von selbst und muss im Projekt eingeplant werden. Der Aufwand zahlt sich allerdings spätestens in der Realisierung aus.

Denkanstoß: Stellen Sie sich einen Toaster vor, der gleichzeitig Spiegeleier braten kann. Welche Zielgruppen erreichen Sie mit diesem Produkt:

a.) Toast-Liebhaber

b.) Spiegeleier-Liebhaber

c.) Beide Zielgruppen: Sowohl Toast- als auch Spiegeleier-Liebhaber

d.) Die Schnittmenge: Jene Leute, die bevorzugt Toast zusammen mit Spiegelei zum Frühstück genießen.

Bonusfrage: Wie wird idealerweise die Dauer der Toast-Zeit auf die Dauer der Spiegelei-Bratzeit abgestimmt?

Der obige Denkanstoß soll einen weiteren Aspekt verdeutlichen: Produkte werden häufig mit vielen Features ausgestattet, um möglichst viele Käufer anzusprechen. Mehr Funktionsvielfalt muss aber in der Regel durch eine höhere Komplexität in der Bedienung erkauft werden. Diese zusätzliche Komplexität darf den Nutzen aus Benutzersicht nicht überschreiten, andernfalls akzeptieren die Benutzer das Produkt nicht oder weichen auf Konkurrenzprodukte aus. Die Zielgruppe der potenziellen Benutzer wird damit nicht größer, sondern kleiner. Konsequentes Usability Engineering zeigt solche Zielkonflikte schon zu Beginn der Produktentwicklung auf.

Anwendungsgebiete für Usability Engineering

Gute Usability spielt überall dort eine Rolle, wo Benutzer mit interaktiven technischen Systemen zu tun haben und damit in irgendeiner Form eine Benutzerschnittstelle zum Einsatz kommt. Dies umfasst Software am Arbeitsplatz ebenso wie Produkte, die in der Freizeit verwendet werden. Der Fokus liegt auf Systemen mit grafischer Benutzeroberfläche (GUI), doch auch Sprachdialoge oder physische Bedienelemente können bezüglich Usability optimiert werden.

Die Entwicklung der Benutzerschnittstelle wird ein zunehmend wichtiger Bestandteil im Software Engineering. Das in diesem Buch beschriebene Vorgehen fokussiert auf die Entwicklung von Software-Anwendungen sowie Produkten mit softwarebasierten Benutzerschnittstellen. Aufgrund der vielen Freiheitsgrade und der Komplexität, die moderne Software-Entwicklungen beinhalten, sehen wir hier die größte Wirkung für benutzerorientierte Vorgehensweisen. Die vorgestellten Methoden und Prinzipien lassen sich jedoch ohne Weiteres auch für die Optimierung von Hardware-Produkten, Dienstleistungsangeboten oder Arbeitsprozessen im weiteren Sinne anwenden.

Symptome für ungenügende Usability

Denkanstoß: *Entwickelt Ihr Unternehmen selbst Software oder Produkte? Wie stellen Sie heute sicher, dass Ihre Produkte gut benutzbar sind? Sind Sie zufrieden mit der erreichten Usability?*

Der obige Denkanstoß klingt banal. Doch woher wissen Sie, wie gut die Benutzbarkeit Ihrer Produkte ist? Haben Sie entsprechende Rückmeldungen oder haben Sie diese gemessen? Es ist nicht immer ganz einfach, die Symptome schlechter Usability auf die Ursache zurückzuführen. Entsprechend verstärkt man unter Umständen den Marketingaufwand, statt Usability-Maßnahmen zu ergreifen. Die folgende Liste zeigt einige Symptome, die von ungenügender Usability herrühren können:

- Die Mitarbeiter arbeiten mit den Systemen nicht so schnell wie erhofft.
- Die Einarbeitung und die Schulung von Benutzern nimmt viel Zeit in Anspruch.
- Die Qualität der geleisteten Arbeit sinkt merklich.
- Die Hotline ist überlastet.
- Mitarbeiter minimieren die Tätigkeit am System. Arbeitsschritte werden auf andere Art und Weise gelöst.
- Prozessvorgaben werden umgangen und Sicherheitsmaßnahmen ignoriert.
- Es gibt immer wieder Fälle, in welchen „Benutzerfehler“ die Ursache von Schäden (Unfälle, Datenverluste, kommerzielle Schäden) sind.

Ein Blick in die Vergangenheit

Um Usability Engineering als Fachdisziplin einzuordnen, lohnt sich ein Blick zurück. Ohne Anspruch auf einen vollständigen geschichtlichen Abriss möchten wir hier einige Meilensteine und Personen aufführen, die maßgeblich zur Entstehung und Verbreitung des Gebietes beigetragen haben:

- Im 15. Jahrhundert stellt Leonardo da Vinci die Kenntnis des Menschen in den Mittelpunkt für die Entwicklung neuer Technologien. Sein Gedankengut sollte Wissenschaft und Technik nachhaltig beeinflussen.
- In den 40er-Jahren des letzten Jahrhunderts investiert vor allem das amerikanische Militär in die Optimierung der Mensch-Maschine-Schnittstelle komplexer Systeme. **Human Factors**, das Fachgebiet zur Erforschung menschlicher Einflussgrößen bei der Anwendung von Technologien, entsteht.
- 1957 erscheint die erste Ausgabe der Fachzeitschrift *Ergonomics*, welche die internationale Verbreitung der **Ergonomie** als Wissenschaft zur Erforschung der Beziehungen zwischen dem Menschen und seiner Arbeit auslöst.
- 1970 gründet Brian Shackel in England das Forschungsinstitut HUSAT (*Human Sciences and Advanced Technology*). Die Erforschung der Kommunikation zwischen Mensch und Computer (**Human Computer Interaction** oder kurz *HCI*) wird zur anerkannten wissenschaftlichen Disziplin.
- Mitte der 1980er-Jahre erlebt die systematische Untersuchung der Mensch-Computer-Interaktion durch die Disziplin der **Software-Ergonomie** einen Aufschwung. Insbesondere im deutschen Sprachraum erscheint eine Fülle von Veröffentlichungen, die das Gebiet prägen.
- 1988 veröffentlicht Donald Norman sein heute als Klassiker geltendes Werk *The Psychology of Everyday Things* [Norman 88]. Das Buch verdeutlicht auf eindrucksvolle Weise die Relevanz psychologischer Faktoren bei der Entwicklung von technischen Systemen.
- 1993 erscheint das Buch *Usability Engineering* von Jakob Nielsen [Nielsen 93]. Es beschreibt die Anwendung von Usability-Methoden in einem systematischen Prozess und gilt als Wegbereiter für benutzerorientierte Vorgehensweisen.

- Im Internet-Boom Ende der 1990er-Jahre setzen viele Unternehmen auf das World Wide Web. Die Nachfrage für die Erstellung benutzerfreundlicher Websites und Anwendungen steigt explosionsartig. *Web Usability* wird zum Schlagwort.
- Im neuen Jahrtausend führt die zunehmende Digitalisierung von Inhalten wie Musik, Foto, Video und die für jedermann erschwinglichen Geräte zu einer weiteren Veränderung. Der Computer wird zum Arbeits-, Kommunikations- und Unterhaltungsmittel und damit zum Alltagsgegenstand. Neue Technologien mit revolutionären Konzepten für die Mensch-Computer-Interaktion werden alltags-tauglich und durchdringen den Markt. Usability wird zum neuen Differenzierungsfaktor für Unternehmen und ihre Produkte.

Usability Engineering im Entwicklungsprozess

Ein Vorgehen für Usability Engineering zu beschreiben, das auf all die verschiedenen Situationen passt, die Sie antreffen werden, ist unserer Ansicht nach nicht möglich. Sie werden auch kein „Kochbuch“ für die Anwendung von Usability-Methoden zu lesen bekommen. Die folgenden Kapitel sollen vielmehr eine Hilfestellung geben, wie Usability Engineering in der Praxis der Software- und Produktentwicklung betrieben werden kann, wie die wichtigsten Usability-Methoden grundsätzlich ablaufen, wie sie eingeplant werden können und worauf Sie achten sollten.

Dieses Kapitel gibt eine Übersicht über den Zusammenhang der verschiedenen Usability-Methoden und deren Integration in gängige Software-Engineering-Vorgehensmodelle. Im nächsten Kapitel werden die einzelnen Methoden näher beschrieben.

Software Engineering: Die vergessenen Benutzer?

Gemäß einer viel zitierten Studie der Standish Group sind *unvollständige Anforderungen* und *mangelnde Einbeziehung der Benutzer* die zwei führenden Gründe, warum Software-Projekte scheitern [Standish Group 1994-2009]. Aktuelle Software-Engineering-Vorgehensmodelle nehmen für sich in Anspruch, hier Abhilfe zu schaffen. Tätigkeiten im Bereich Anforderungsanalyse und -management haben in den meisten neueren Prozessmodellen einen hohen Stellenwert.

Der *Rational Unified Process (RUP)*, ein etablierter Software-Entwicklungsprozess, definiert beispielsweise **Requirements Engineering** als eine von sechs primären Disziplinen im Software Engineering. Sie umfasst jene Aktivitäten, bei denen es um das Erfassen, Dokumentieren und Verwalten von Anforderungen der verschiedenen Interessengruppen geht (siehe auch Abschnitt „Requirements Engineering“, S. 13). Als eine der wichtigsten Interessengruppen spielen die Benutzer dabei eine zentrale Rolle. Für Usability-Engineering-Aktivitäten sind im RUP einige wesentliche Rahmenbedingungen erfüllt:

- Die Bedürfnisse der Benutzer werden im Rahmen der Anforderungsanalyse erhoben.

- Die Anforderungen aus Benutzersicht werden modelliert und in der Spezifikation für die Entwicklung festgehalten.
- Eine iterative Vorgehensweise erlaubt die Überprüfung und Verfeinerung von Ergebnissen.
- Einige Best Practices aus dem Usability Engineering, wie User Interface Storyboards zur Darstellung von Dialogabläufen oder GUI-Prototypen, werden dabei explizit erwähnt.

Es wäre somit zu erwarten, dass in Software-Projekten generell und in der Entwicklung von Benutzerschnittstellen im Speziellen eine systematische Einbeziehung der Benutzer und ihrer Anforderungen stattfinden müsste.

Ist das Problem damit gelöst? Die Praxis spricht eine andere Sprache. Gemäß einer repräsentativen Befragung bei Software-Entwicklern in der Schweiz [Vukelja, Müller & Opwis 07] wird das User Interface (UI) in 47 % der Fälle vom Entwickler allein, d. h. ohne fremde Hilfe erstellt. In nur 27 % der Fälle wurden die späteren Benutzer für die Anforderungen und das GUI Design einbezogen. Aus den Antworten von 92 % der Entwickler konnte zudem geschlossen werden, dass sie keine oder nur wenige Kenntnisse im Bereich Mensch-Computer-Interaktion besitzen. Wenn man sich nochmals die oben genannten Gründe für das Scheitern von Projekten vergegenwärtigt, muss dieses Ergebnis überraschen.

Ein Grund dafür mag in Folgendem liegen: Obwohl die Berücksichtigung der Benutzeranforderungen in aktuellen Software-Engineering-Vorgehensmodellen wie dem RUP als wichtig erachtet wird, fehlen für die tatsächliche Einbeziehung von „Endbenutzern“ entsprechende Aktivitätsbeschreibungen, Methoden und Techniken weitgehend. Damit fehlen auch Erfahrungswerte für die Projektplanung. Das Resultat ist, dass viele Projekte eine Benutzereinbeziehung als optional oder als zusätzlichen Aufwand betrachten und ohne entsprechende Aktivitäten vorgehen.

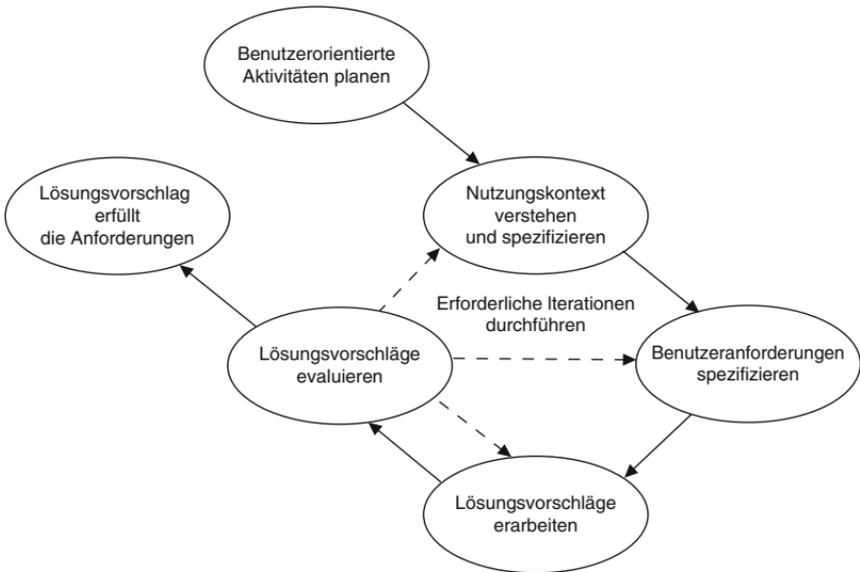
Benutzerorientierte Modelle

Auf der anderen Seite existieren seit gut zehn Jahren benutzerorientierte Vorgehensmodelle, welche die Anwendung von Usability-Methoden im Entwicklungsprozess beschreiben. Die Disziplin **User Centered Design** (UCD) beschäftigt sich damit, wie Benutzer systematisch in die Entwicklung von Systemen und Produkten einbezogen werden können. An dieser Stelle sei auf die hervorragenden,

durchgängig benutzerorientierten Vorgehensbeschreibungen von namhaften Vertretern des Fachgebiets verwiesen:

- *Usability Engineering Lifecycle* von Deborah Mayhew [Mayhew 99]
- *Contextual Design* von Hugh Beyer und Karen Holtzblatt [Beyer & Holtzblatt 98]
- *Goal Directed Design* von Alan Cooper [Cooper & Reimann 07]
- *Interaction Design Lifecycle* von Jenny Preece [Sharp, Rogers & Preece 07]

Ein Prozessmodell für die *benutzerorientierte Gestaltung interaktiver Systeme* wurde sogar in einer ISO-Norm verankert [DIN EN ISO 13407] bzw. neu [ISO/DIS 9241-210]. In der folgenden Abbildung sind die wichtigsten Prozessschritte dargestellt. Eine genaue Beschreibung dieser Schritte ist in ISO 9241-210 enthalten.



Wer nach diesen Ansätzen vorgeht, wird bessere Software und Produkte produzieren – kein Zweifel!

Eines ist indessen nicht von der Hand zu weisen: Viele Autoren scheinen ihre Vorgehensmodelle weitgehend für Usability-Fachleute zu schreiben. Es entsteht unweigerlich das Gefühl, dass diese Ansätze als aufwändige, eigenständige Disziplinen gehandelt werden: „Vergesst eure alten Vorgehensweisen und nehmt unseren Prozess.“

Eine Integration in gängige Software-Engineering-Prozessmodelle fehlt weitgehend. Diese Abgrenzungshaltung mag dazu beitragen, dass Usability Engineering von Auftraggebern und Projektbeteiligten oft als zeit- und kostenintensiver Zusatz zum bestehenden Vorgehen betrachtet wird.

Wenn Sie nun aber ein Projekt nach einem bestehenden Vorgehensmodell oder einem gegebenen Entwicklungsprozess abwickeln müssen, weil Ihr Unternehmen dies vorschreibt oder weil Sie darin ganz einfach viele nützliche Hilfsmittel finden, wie können Sie benutzerorientierte Methoden integrieren? Sollen Sie nun Geschäftsprozessmodellierung oder Benutzerbeobachtungen in Auftrag geben, um die notwendigen Arbeitsschritte zu analysieren? Verwenden Sie besser Szenarien oder Use Cases für Ihre Spezifikation? Wann testen Sie Ihre neue Lösung am besten mit Benutzern – und mit wie vielen? Die folgenden Kapitel sollen Sie bei solchen und ähnlichen Fragen unterstützen.

Das Beste aus beiden statt zwei Welten

Die Erfahrung aus einer Vielzahl von Projekten hat uns gezeigt, dass sich Usability Engineering nahtlos in gängige Software-Entwicklungsprozesse integrieren lässt und sich dadurch viele Vorteile bieten. So hat sich beispielsweise die Anwendung von Usability-Methoden im *Requirements Engineering* in der Praxis als äußerst erfolgreich erwiesen. Erstens werden die Anforderungen der Benutzer systematisch in die Analyse einbezogen und zweitens kann sichergestellt werden, dass diese Anforderungen auch ihren Weg in die Realisierung finden. Das Ergebnis ist eine Lösung, die sich an ihrem tatsächlichen Verwendungszweck ausrichtet.

Eine solche integrierte Sicht bedingt, dass Vorgehensbeschreibungen nicht einfach unreflektiert angewendet werden, sondern dass die zugrunde liegenden Prinzipien beider Welten verstanden und berücksichtigt werden.

Hintergrund: Requirements Engineering

Requirements Engineering befasst sich damit, die Bedürfnisse der Benutzer, Auftraggeber und weiterer Interessengruppen (*Stakeholder*) hinreichend aufzuarbeiten, zu verwalten und zu kommunizieren, damit das Projektteam eine passende Lösung erstellen kann. Wesentlich für den Projekterfolg ist eine von allen Interessengruppen vereinbarte und getragene Zielvorstellung. Aufbauend darauf kann der Analyst die Rahmenbedingungen und Qualitätsanforderungen

erarbeiten sowie den funktionalen Umfang und das Verhalten der neuen Lösung modellieren. Eine kurze und gute Übersicht über die Thematik bietet das Buch *Anforderungsmanagement kompakt* [Dörnemann & Meyer 03].

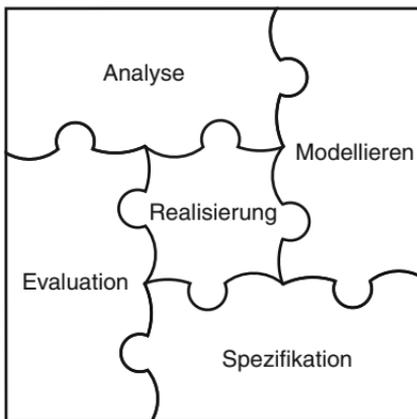
Usability-Methoden im Zusammenhang

Hinweis: Dieser Abschnitt zeigt die Integration von Usability-Methoden und gängigen Software-Engineering-Vorgehensweisen. Er ist als Übersicht für den erfahrenen Leser gedacht. Dem Einsteiger in die Materie empfehlen wir, sich im folgenden Kapitel („Die 7±2 wichtigsten Usability-Methoden“, S. 21) zunächst mit den einzelnen Methoden und Konzepten vertraut zu machen.

Die Tätigkeiten in einer benutzerorientierten Software- oder Produktentwicklung lassen sich vereinfacht in fünf Aufgabenbereiche zusammenfassen:

- *Analyse*: Benutzer und Kontext verstehen
- *Modellieren*: Entwurf und Optimierung einer passenden Lösung
- *Spezifikation*: Die neue Lösung für die Entwicklung festhalten
- *Realisierung*: Unterstützung bei der Implementierung der Lösung
- *Evaluation*: Resultate mit Benutzern überprüfen

Die Einteilung in Aufgabenbereiche erleichtert das Verständnis von Ziel und Zweck der eingesetzten Methoden und der gewünschten Arbeitsergebnisse. Die folgende Abbildung zeigt die Aufgabenbereiche einer benutzerorientierten Entwicklung in der Übersicht:



Diese Aufgabenbereiche sind nicht im Sinne eines zeitlichen Ablaufs zu verstehen. So kann es beispielsweise zielführend sein, zunächst eine Evaluation eines bestehenden Systems vorzunehmen, bevor die Modellierung einer neuen Lösung angegangen wird. Oder es kann ein erstes Modell erstellt werden, das der Analyse dient. In der Praxis werden diese Aufgaben oft auch wiederholend (iterativ) oder sogar parallel durchgeführt. Im Kapitel

„Usability im Griff: Planung“ (S. 77) werden diese zeitlichen Zusammenhänge näher beschrieben.

Im Folgenden wird für jeden Aufgabenbereich aufgezeigt, welche Ansätze und Methoden einerseits aus den klassischen Software-Engineering-Vorgehensmodellen und andererseits aus benutzerorientierten Prozessmodellen zur Verfügung stehen und wie diese in einem integrierten Vorgehen zusammenspielen. Die Details der einzelnen Usability-Methoden werden im nächsten Kapitel beschrieben.

Aufgabenbereich 1: Analyse von Benutzern und Kontext

Am Anfang eines neuen Projekts steht der Projektleiter vor der Herausforderung, eine neue Lösung zu planen, von der er noch nicht genau wissen kann, wofür sie eingesetzt wird, was sie bieten soll und wer die Benutzer sein werden.

In der **Geschäftsprozessanalyse bzw. -modellierung** (*Business Analysis, Business Modeling*) werden die Prozesse des Unternehmens analysiert, in welche die neue Lösung eingebettet werden soll. Dabei muss auch der Tatsache Rechnung getragen werden, dass sich diese Prozesse in der Regel durch die Einführung der Lösung verändern. Es würde den Rahmen dieses Buches sprengen, die Tätigkeiten der Analyse und Modellierung im Detail zu beschreiben. Wichtig ist an dieser Stelle, dass die Geschäftsprozesse wesentliche Rahmenbedingungen für die neue Lösung definieren, zum Beispiel, welche Stellen und Prozesse im Unternehmen betroffen sind, welche Aufgaben und Tätigkeiten von diesen Stellen ausgeführt werden und wie der offizielle Informationsfluss und die Zusammenarbeit zwischen den verschiedenen Stellen stattfinden. Basierend auf diesen Informationen lässt sich aus Geschäftssicht definieren, für welche Aufgaben das neue System eingesetzt werden soll.

Neben den formellen Geschäftsprozessen ist ein fundiertes Verständnis über die Benutzer und die tatsächliche Anwendung der neuen Lösung im Geschäftsalltag eine notwendige Voraussetzung. Usability Engineering komplettiert die formellen Prozesse aus der Geschäftsprozessmodellierung mit den konkreten Details der täglichen Arbeit und füllt die abstrakten Unternehmens-Rollen mit den Eigenschaften und Möglichkeiten der Menschen dahinter.

Contextual Inquiry (siehe Abschnitt „Contextual Inquiry“, S. 22) soll in diesem Buch als stellvertretendes Beispiel für weitere Techniken stehen, die von Beobachtungen über Interviews und moderierte

Gesprächsgruppen (*Focus Groups*) bis zu strukturierten Aufgabenanalysen reichen. Das Ziel dieser Methoden ist es, die Bedürfnisse der Benutzer sowie die Hintergründe zu verstehen. In Beobachtungen und Befragungen der Benutzer vor Ort werden die konkreten Arbeitsabläufe und Verhaltensmuster sowie die Umgebung der Anwendung analysiert, ausgewertet und dokumentiert. Die Resultate hält der Analyst in Form von grafischen Modellen oder natürlichsprachlichen Beschreibungen fest.

Die Erarbeitung von Anforderungen der verschiedenen Interessengruppen ist eine zentrale Aufgabe im Requirements Engineering. Die Analysetätigkeiten umfassen beispielsweise Interviews, moderierte Workshops sowie die Analyse von Altsystemen und Dokumentationen. Usability-Methoden dienen zur Erarbeitung von Anforderungen aus Benutzersicht und ergänzen damit die Techniken im Requirements Engineering.

Aufgabenbereich 2: Modellieren einer passenden Lösung

Selbst bei kleineren Anwendungen ist es praktisch unmöglich, auf Anhieb ein System zu entwerfen, das allen Anforderungen genügt. Es ist notwendig, das neue System aus verschiedenen Sichten zu modellieren und Schritt für Schritt zu präzisieren. In mehreren Zyklen werden Entwürfe erstellt und Feedback eingeholt. Der Analyst modelliert beispielsweise die Einbettung in die Geschäftsprozesse, die Arbeitsweise der Benutzer mit dem neuen System sowie die Funktionalität und das Verhalten des Systems.

Die Erkenntnisse aus der Analyse von Benutzern und Kontext werden in Personas und Szenarien umgesetzt (siehe Abschnitt „Personas und Szenarien“, S. 28). **Personas** stellen prototypische Benutzerprofile dar, während **Szenarien** die Arbeit mit dem neuen System aus Benutzersicht beschreiben. Personas und Szenarien bringen die im Contextual Inquiry erarbeiteten Ergebnisse auf den Punkt und dienen als Grundlage für die Entwicklung. Sie werden in Anforderungsdokumenten festgehalten.

Personas und Szenarien liefern eine gute Basis für den Entwurf **eines Use-Case-Modells** (siehe Abschnitt „Use Cases“, S. 48). Dabei wird zuerst eine Übersicht über den funktionalen Umfang der neuen Lösung erarbeitet und in einem zweiten Schritt das Verhalten des Systems gemäß den Qualitätsanforderungen und Rahmenbedingungen beschrieben. Als Modellierungselemente dienen **Akteure**

(Actors), die in **Anwendungsfällen** (*Use Cases*) mit dem System interagieren.

Ein genaues Verständnis der Objekte und Daten des Fachbereichs, welche im System repräsentiert werden sollen, ist eine wichtige Voraussetzung für die Erstellung der Benutzerschnittstelle. Diese Konzepte und Zusammenhänge können im *Domänenmodell* abgebildet werden. Ein *Glossar* ist dabei ein einfaches Hilfsmittel, um Begriffe zu definieren und ein gemeinsames Verständnis im Projekt zu erreichen.

Mit **Storyboards** (siehe Abschnitt „Storyboards“, S. 36) hat der Analyst die Möglichkeit, Lösungsideen anschaulich darzustellen, um beispielsweise Feedback von Benutzern und Führungskräften zu erhalten. Storyboards visualisieren und detaillieren die Anwendung aus Benutzersicht, indem sie einen ersten Eindruck über die Abläufe der Benutzerschnittstelle im vorgesehenen Kontext geben.

Um das Verhalten des Systems weiter zu verfeinern und die Ablaufschritte aus Benutzersicht in Aktion darzustellen, werden im **User Interface Prototyping** (siehe Abschnitt „UI Prototyping“, S. 40) erste Dialogschritte mit möglichst einfachen Mitteln umgesetzt und von ausgewählten Benutzern und weiteren Stakeholdern geprüft. Die so erstellten Prototypen vermitteln den Beteiligten erstmals einen Eindruck der laufenden Anwendung und dienen als einfach verständliche und gemeinsame Sprache zwischen Benutzern, Auftraggebern und Entwicklern. Die in Anwendungsfällen textuell oder mit grafischen Modellen formulierten Anforderungen werden dadurch greifbar und vorstellbar. Dank der erhöhten Realitätsnähe kommen bisher unentdeckte Anforderungen zum Vorschein. Ein wichtiges Ergebnis ist das User-Interface-Konzept (siehe Abschnitt „Die Benutzerschnittstelle konzipieren“, S. 42), das die Eckpunkte für das zu erstellende User Interface festlegt.

Aufgabenbereich 3: Spezifizieren für die Entwicklung

Wenn genügend Klarheit über die vorgesehene Lösung herrscht, muss das zu erstellende System für die Entwicklung spezifiziert werden. Auftraggeber und Hersteller legen mit der Anforderungsspezifikation den fachlichen Inhalt für die Vertragserfüllung fest. Use-Case-Spezifikationen beschreiben die funktionalen Abläufe des Systems (siehe Abschnitt „Use Cases“, S. 48). Neben weiteren funktionalen Anforderungen werden auch alle Rahmenbedingungen aufgenommen

(siehe Abschnitt „Funktionale und nicht-funktionale Anforderungen“, S. 51). Die Anforderungen an die neue Lösung werden in strukturierten Dokumenten oder auch in speziellen Werkzeugen festgehalten.

Zwischen den Tätigkeiten für das Modellieren der neuen Lösung und deren Spezifikation besteht ein fließender Übergang. Ausgewählte Ergebnisse aus der Modellierung dienen auch der Spezifikation. So können Szenarien, Storyboards oder UI-Prototypen auch Teil der Anforderungsspezifikation sein und wesentlich zur Verständlichkeit, Vollständigkeit und zur Präzisierung der Anforderungen für die Realisierung beitragen. Die **Spezifikation** besteht schließlich aus einem Konglomerat von formalen Anforderungsnotationen und natürlichsprachlichen Beschreibungen wie dem Use-Case-Modell, Use-Case-Spezifikationen, Ablaufdiagrammen, dem Domänenmodell, Anforderungssätzen, nicht-funktionalen Anforderungen und zusätzlichen Ergebnissen wie Storyboards und UI-Prototypen.

Aufgabenbereich 4: Unterstützung der Realisierung

Für die Realisierung der Lösung muss die Spezifikation in ein technisches Design umgesetzt, eine Software-Architektur entworfen und implementiert werden. Vorgehensmodelle in der Software- oder Produktentwicklung setzen hier ihren Schwerpunkt. In der Realisierung unterstützen **Usability Guidelines** und **Styleguides** die Entwicklung und helfen, ein konsistentes und regelkonformes User Interface Design zu erreichen (siehe Abschnitt „Usability Guidelines und Styleguides“, S. 54). User-Interface-Prototypen bilden eine wertvolle Grundlage für die Entwicklung (siehe Abschnitt „UI Prototyping“, S. 40). Moderne Software-Entwicklungsprozesse und agile Vorgehensweisen unterstützen dabei einen iterativen Prozess, bestehend aus der Erstellung von Lösungsvarianten und Einarbeiten von Feedback in kurzen Zyklen. Diese Denkhaltung lässt sich ausgezeichnet mit benutzerorientierten Vorgehensweisen vereinbaren.

Aufgabenbereich 5: Evaluation der Resultate

Ein wesentlicher Aufgabenbereich einer benutzerorientierten Software- oder Produktentwicklung besteht darin, die erstellten Resultate mit Benutzern zu überprüfen und zu optimieren. Dabei kann es sich bereits um ein realisiertes System oder auch erst um einen Prototypen handeln. Interessanterweise bieten die gängigen Software-

Engineering-Vorgehensmodelle hier relativ wenige Hilfsmittel an. Im Wesentlichen beschränken sie sich auf formale Reviews, Stellungnahmen und Abnahmetests durch den Auftraggeber.

Hingegen gibt es zahlreiche Usability-Methoden zur Evaluation eines Systems. In einem formalen **Usability-Test** werden Benutzer in einem Usability Lab dabei beobachtet, wie sie mit einer neuen Anwendung oder einem Prototypen arbeiten (siehe Abschnitt „Usability Testing“, S. 60). Die Probleme werden dokumentiert und daraus Verbesserungsmaßnahmen erarbeitet. Usability-Tests können auch im Rahmen eines Abnahmetests zur Erfüllung der Anforderungen eingesetzt werden. **Usability Walkthroughs** sind weniger formal und eignen sich zur Überprüfung und Optimierung von ersten Prototypen früh im Prozess. Mithilfe von **Usability-Fragebögen** kann von einer größeren Anzahl von Benutzern Feedback zur Benutzbarkeit einer neuen Lösung eingeholt werden. Schließlich besteht die Möglichkeit, Benutzerschnittstellen anhand von Checklisten oder durch Experten prüfen zu lassen.

Zusammenfassung der Methoden

Die folgende Tabelle zeigt die Integration der Methoden aus Software Engineering und benutzerorientierten Prozessmodellen nochmals als Übersicht. Die kursiv gedruckten Methoden werden im nächsten Kapitel näher beschrieben.

Aufgabenbereich	SE-Vorgehensmodelle	Benutzerorientierte Prozessmodelle
Analyse	Business Analysis Business Modeling Stakeholder-Interviews Moderierte Workshops Analyse von Altsystemen	<i>Contextual Inquiry</i> Beobachtungen Interviews Focus Groups Aufgabenanalysen <i>Fragebögen</i>
Modellieren	Business Modeling <i>Use-Case-Modell</i> <i>Use Cases</i> Domänenmodell Glossar	<i>Personas und Szenarien</i> <i>Storyboards</i> <i>UI Prototyping, Mock-ups</i> <i>User-Interface-Konzept</i>
Spezifikation	<i>Use-Case-Modell</i> <i>Use-Case-Spezifikationen</i> Nicht-funkt. Anforderungen Ablaufdiagramme Domänenmodell Anforderungssätze	<i>Szenarien</i> <i>Storyboards</i> <i>UI-Prototypen</i> <i>Styleguides</i>

Aufgabenbereich	SE-Vorgehensmodelle	Benutzerorientierte Prozessmodelle
Realisierung	Technisches Design SW-Architektur Implementierung	<i>Usability Guidelines</i> <i>Styleguides</i> <i>UI-Prototypen</i>
Evaluation	Formale Reviews Stellungnahmen Funktionales Testen Abnahmetests	<i>Usability Testing</i> Walkthroughs <i>Fragebögen</i> Checklisten und Heuristiken Experten-Reviews

Die 7±2 wichtigsten Usability-Methoden

Im vorigen Kapitel haben wir argumentiert, dass sich Usability Engineering nahtlos in bestehende Software-Engineering-Ansätze integrieren lässt, und Ihnen einige Methoden im Zusammenhang aufgezeigt. In diesem Kapitel möchten wir Ihnen acht* wichtige Usability-Methoden näher vorstellen. Wir sind davon überzeugt, dass diese Auswahl einem Projekt eine umfassende Werkzeugpalette zur Verfügung stellt, um in unterschiedlichen Situationen gezielt benutzbare Software entwickeln zu können. Die folgende Tabelle fasst diese acht Methoden und ihren primären Zweck zusammen:

Methode	Zweck
Contextual Inquiry	Analyse der Benutzer und des Einsatzumfelds des neuen Systems
Personas und Szenarien	Modellieren der unterschiedlichen Benutzergruppen und der Anwendung aus Benutzersicht
Storyboards	Kommunizieren von ausgewählten Abläufen mit dem neuen System
User Interface Prototyping	Klären der Anforderungen, Konzipieren und Optimieren der Benutzerschnittstelle
Use Cases	Spezifizieren der funktionalen Anforderungen für die Entwicklung
Guidelines und Styleguides	Definieren der Gestaltungsrichtlinien
Usability Testing	Beurteilen des neuen Systems durch Benutzer

* Eine Randnotiz für den interessierten Leser: Bei der Zahl im Titel handelt es sich um nichts anderes als die oft zitierte „magische Zahl 7±2“. Der Psychologe George A. Miller veröffentlichte 1956 eine Studie über die Limitationen der menschlichen Informationsverarbeitung [Miller 56]. Das menschliche Gehirn ist demnach in der Lage, im Durchschnitt maximal 5 bis 9 gleichwertige Sinnesreize zu beurteilen. Diese Grenze konnte durch verschiedene Experimente, unter anderem mit Tonhöhen, Lautstärken, visuellen Stimuli usw., nachgewiesen werden. Dieselbe Zahl wurde auch in anderen Studien gefunden: 7±2 entspricht etwa der Anzahl Informationen, die ein Mensch gleichzeitig im Kurzzeitgedächtnis behalten kann. 7±2 ist die Anzahl Objekte, welche die menschliche Aufmerksamkeit umfasst. Missverständnis, Zufall oder Gesetzmäßigkeit? Auf jeden Fall galt die *magische Zahl 7±2* fortan als eine Art fundamentale Konstante. In der Software-Ergonomie wird argumentiert, dass aufgrund der Limitationen der menschlichen Informationsverarbeitung eine effiziente Auswahl nur etwa 5 bis 9 gleichartige Elemente enthalten sollte.

Methoden

Fragebögen

Zweck

Sammeln aussagekräftiger Zahlen zur Analyse von Benutzern und Kontext oder zur Beurteilung eines Systems oder Prototyps

Facetten der Arbeit: Contextual Inquiry

Es locken heiße Features, coole Technologien und verführerisches Sparpotenzial. Sie brauchen nur zuzugreifen. Scheinbar ist die seit über zwanzig Jahren akute Software-Krise nur eine Frage der richtigen Technologie und heute dank SOA und MDA gelöst (ersetzen Sie die Abkürzungen mit beliebigen Schlagworten der aktuellen IT-Marketing-Maschinerie).

Contextual Inquiry geht die Software-Krise nicht durch neue Technologien an, sondern durch ein fundiertes Verständnis der künftigen Benutzer, ihren Tätigkeiten und Bedürfnisse. Contextual Inquiry lässt sich mit „Erhebung im Umfeld der Benutzer“ übersetzen. Der Analyst untersucht die Bedürfnisse der Anwender, indem er diese bei ihren Tätigkeiten beobachtet und sie dazu befragt.

Von Beobachtung und Befragung zu Bedürfnissen

Eine neue Software soll Versicherungsberater beim Erstellen von Offerten unterstützen. Das Projektteam besucht die Benutzer, um die Arbeitsweise der Berater zu verstehen und eine passende Software zu erarbeiten. Besonders relevant ist das eigentliche Beratungsgespräch, da die Software gemäß Projektauftrag die Qualität der Beratung verbessern soll. Die Analysten nehmen an einigen Beratungsgesprächen teil. Sie beobachten das Gespräch und befragen zudem den Berater über den Verlauf. Die Beobachtung ermöglicht es den Analysten, wichtige Informationen aufzunehmen, die in einer reinen Befragung nicht auftauchen würden. Die Analysten zeichnen im Detail auf, welche Informationen tatsächlich relevant sind und wie die Berater daraus auf geeignete Versicherungsprodukte schließen. Die Analysten machen sich ebenfalls ein Bild von einigen Software-Anwendungen, die von den Beratern eingesetzt werden, beispielsweise um Berechnungen durchzuführen, Informationen abzulegen, Briefe zu schreiben etc. Diese detaillierten Informationen erlauben es dem Projektteam, ein zu den Aufgaben der Benutzer passendes System zu entwerfen.

Die psychologische Forschung bestätigt, dass Menschen viel Wissen, das sie in einer bestimmten Situation anwenden, nicht einfach abrufen können. In einem Interview ist solches Wissen nur schwer erfassbar. Die Kombination von Beobachtung und Befragung ermöglicht einerseits, das wirkliche Geschehen im Detail zu erfassen, und andererseits, die Gründe und Zusammenhänge dahinter zu durchleuchten. Diese Informationen sind wertvoll, um den notwendigen Informationsgehalt, die Benutzerführung und die Funktionalität des geplanten Systems abzuleiten.

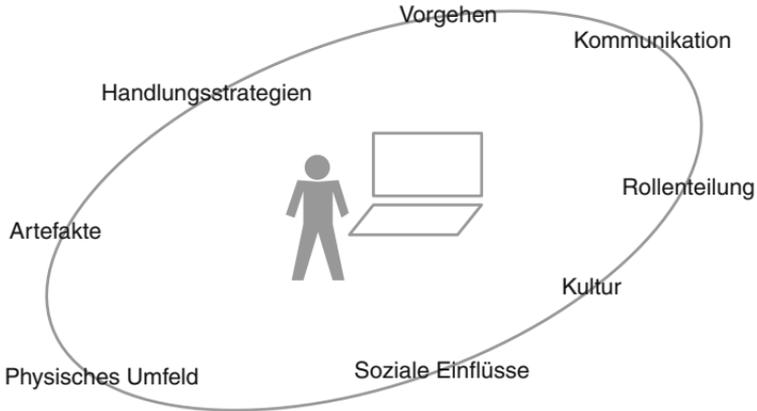
Fragestellung

Mit Contextual Inquiry sollen ausgewählte Fragestellungen beantwortet werden. Lohnenswerte Fragen zielen auf den Einsatz und das Umfeld von heute benutzten Produkten ab. Für die Weiterentwicklung eines Navigationsgerätes im Auto kann es beispielsweise aufschlussreich sein, wie die Rollenverteilung zwischen Beifahrer und Fahrer auf einer Urlaubsreise ist.

Bevor ein Analyst mit Benutzern spricht, stellt sich das Projektteam die Frage, was in Bezug auf das neue System wichtig zu wissen ist. Natürlich ergeben sich auch Fragestellungen, die mit Contextual Inquiry nicht beantwortet werden können: Beispielsweise wie viel ein Kunde für ein Produkt bezahlen würde (Marketing und Marktforschung), ob eine bestimmte Technologie geeignet ist (Entwicklung) oder wie ein Geschäftsprozess grundsätzlich ablaufen soll (Geschäftsprozessmodellierung).

Contextual Inquiry fokussiert auf die Tätigkeiten der späteren Benutzer und das Umfeld der Anwendung. Die folgenden fünf Sichten stellen Aspekte dar, die mit der Methode erfasst und dokumentiert werden können:

Die 7±2 wichtigsten Usability-Methoden



Sicht

Rollenteilung und Kommunikation

Handlungsstrategien und Vorgehen

Artefakte

Kulturelle und soziale Einflüsse

Physisches Umfeld

Fragestellung

Typische Rollenverteilungen
 Aufgaben und Verantwortlichkeiten
 Kommunikationsmittel
 Kommunikationszweck und Inhalte
 Vorteile und Probleme der Rollenteilung

Ausführung von Tätigkeiten
 Unterschiedliche Vorgehensweisen
 Stärken und Schwächen
 Häufigkeit, Frequenz, Intensität und Dauer der Durchführung
 Ausnahmesituationen und Fehler, Spezialfälle

Bei der Arbeit benutzte Dokumente, Formulare, Werkzeuge usw.
 Aufbau und Informationsgehalt
 Verwendungszweck
 Anpassung an individuelle Bedürfnisse
 Zweckentfremdete Verwendung
 Vorteile und Probleme bei der Arbeit

Personen, die Einfluss nehmen
 Wirkung von sozialem Druck, Machtausübung
 Verhaltensregeln
 Ziele, Werte und Vorlieben
 Widersprüchliche Einflüsse
 Probleme und Chancen auf kultureller Ebene

Raumaufteilung, Arbeitsplatzgestaltung
 Verfügbare Hilfsmittel
 Wege und Distanzen
 Einfluss auf Kommunikation
 Verbesserungspotenzial

In kaum einem Umfeld ist der gesamte Nutzungskontext in nur einer Interviewrunde erfassbar. Die Erfahrung zeigt, dass die Fragestellungen zu Beginn breit und unscharf sind. Im Verlauf des Projektes kennt das Analyseteam den Kontext immer besser, und die Fragestellung wird konkreter und enger gefasst. Es lohnt sich deshalb, mehrere Iterationen durchzuführen. Die Resultate befruchten jeweils die Fragestellung der nächsten Iteration.

Im Kontext untersuchen

Mit einer ausgewählten Fragestellung stößt der Analyst gezielt in die Welt der Benutzer vor. Die Auswahl der Interviewpartner muss nicht repräsentativ im Sinne der Statistik sein, doch sie sollte ein breites Spektrum an Meinungen und Bedürfnissen abdecken. Es lohnt sich, auf eine gewisse Streuung bezüglich Alter, Geschlecht, Position, Arbeitsort, Erfahrung, Fachwissen, kulturellem Hintergrund und mehr zu achten.

Die Untersuchung findet vor Ort und während der Arbeit statt. Der Analyst beobachtet den Interviewpartner und stellt gezielt Fragen über das Beobachtete. Der Interviewpartner soll die eigene Handlungsweise reflektieren und so angewandtes Expertenwissen aufdecken. Der Analyst und der Interviewpartner diskutieren, ausgehend von einer gerade vorgeführten Arbeitstätigkeit, über Probleme, fachliche Zusammenhänge und Verbesserungsmöglichkeiten. Der Analyst sammelt alles, was im Interview diskutiert wird: ausgefüllte Formulare, Screenshots, Skizzen über interessante fachliche Zusammenhänge, Audioaufnahmen von Gesprächen und mehr.

Bei neuartigen Produktentwicklungen sollten erste Entwürfe, Prototypen oder existierende vergleichbare Produkte in die Interviews einbezogen werden. Je näher die Erhebungssituation an die Realität der geplanten Anwendung heranreicht und je lebensechter der Kontext dieser Anwendung vermittelt werden kann, desto wertvoller sind die Rückmeldungen der Benutzer.

Analysieren der gesammelten Daten

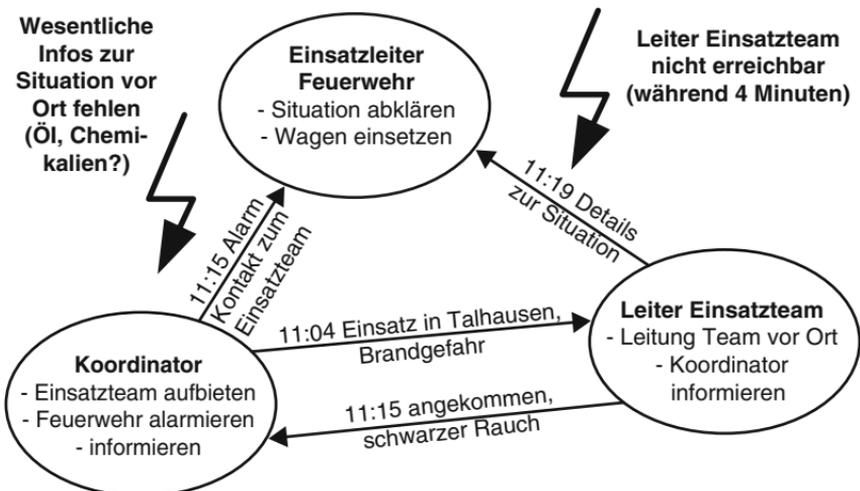
Die Analyse basiert auf den gesammelten Unterlagen, Notizen, Skizzen, Video- und Audioaufnahmen. Es ist von Vorteil, in einem Team aus Analysten und Entwicklern zu arbeiten: Analysten stellen sich andere Fragen und suchen nach anderen Lösungsansätzen als

Die 7±2 wichtigsten Usability-Methoden

Entwickler. Entwickler lernen dabei insbesondere das Arbeitsumfeld kennen. Das Team extrahiert die folgenden Informationen:

- *Ziele und Bedürfnisse der befragten Personen sowie deren Probleme, Werte und Eigenheiten:* Im Abschnitt „Personas und Szenarien“ (S. 28) zeigen wir, wie das Team daraus mittels Personas das Zielpublikum charakterisiert.
- *Aufgaben, Abläufe und Tätigkeiten:* Diese Informationen dienen als Grundlage für die Beschreibung der künftigen Abläufe mit dem neuen System.
- *Schwierigkeiten und erprobte Lösungsansätze der Benutzer mit heutigen Werkzeugen:* Diese Informationen schärfen den Blick für die wesentlichen Bedürfnisse der Benutzer und die dazu passenden Funktionen.
- *Begriffe und Informationen zum Domänenmodell:* Das Team wird hier vor allem in gesammelten Formularen und Dokumenten sowie den bestehenden Software-Anwendungen fündig.

[Beyer & Holtzblatt 98] beschreiben mit **Contextual Design** eine weiterführende Methodik, in welcher die Resultate aus einer Contextual Inquiry in grafischen Modellen festgehalten werden. Zur Untersuchung von Störungsfällen kann beispielsweise die Kommunikation zwischen den beteiligten Personen in einem *Informationsflussmodell* aufgezeichnet werden. Die folgende Abbildung zeigt einen kleinen Ausschnitt eines solchen Modells:



Die grafischen Modelle zeigen die komplexe Kommunikation zwischen den Beteiligten in allen Details und weisen auf Stärken und Schwächen hin.

Geschäftsprozessmodellierung ergänzen

Contextual Inquiry ergänzt die Geschäftsprozessanalyse und -modellierung. Geschäftsprozesse geben die konsolidierten und standardisierten Abläufe in einem Unternehmen wieder. Sie modellieren hingegen nicht die Problemlösungsstrategien, die eine bestimmte Person anwendet, welche Abkürzungen und Optimierungen diese vornimmt und wie sie sich mit den Kollegen abspricht. Geschäftsprozesse sagen auch nichts über das konkrete physische und kulturelle Umfeld aus, in der die Benutzung stattfindet. Diese Lücke schließt Contextual Inquiry und bringt die Aspekte der täglichen Arbeit und der konkreten Anwendung der Geschäftsprozesse in ein Projekt hinein.

Innovation aus dem realen Leben

Innovation in einem Unternehmen entspringt unterschiedlichen Quellen: Neue und bessere Technologien ermöglichen neue Produkte und eröffnen neue Geschäftsfelder. Durch die Optimierung der Geschäftsprozesse werden neue Wege aufgezeigt und das bestehende Potenzial einer Firma besser ausgenutzt. Contextual Inquiry erschließt eine weitere Quelle für Innovation: Die Methode zeigt verbreitete Muster, erprobte Lösungsansätze und ungelöste Probleme in der Arbeitswelt auf – Faktoren, die Potenzial für wirklich nützliche Produkte bieten, welche die Benutzer direkt ansprechen.

Darauf sollten Sie achten

- Behalten Sie während der Analyse die ausgewählte Fragestellung im Auge. So manches Projektteam hat sich schon in der Unzahl der Informationen in der Analyse verloren.
- Halten Sie auch fest, worauf Ihre Erkenntnisse zurückzuführen sind. Irgendwann kommt ein Auftraggeber und will wissen, weshalb ein bestimmtes Feature eingebaut wurde.

In Kürze

Methode	Contextual Inquiry
Resultate	Fundiertes Wissen über Benutzer und Kontext Optimierungspotenzial erkannt und abschätzbar Grundlage zur Beurteilung der Nützlichkeit Anforderungen aus dem Nutzungskontext bekannt

Die 7±2 wichtigsten Usability-Methoden

Vorgehen	Benutzer und ihre Tätigkeiten während der Arbeit beobachten und befragen. Iterationen anhand der offenen Fragen steuern. Optimierungspotenzial identifizieren und Lösungsideen protokollieren.
Aufwand [PT = Personentage]	Hängt stark von der Komplexität des Projekts ab. Klein: 1 Iteration, 1–2 Analysten, 3–6 Interviews: 3–12 PT Mittel: 2 Iterationen, 2 Analysten, 6–12 Interviews: 12–30 PT Groß: mehrere Durchgänge à 2 Iterationen mit 3–5 Analysten und jeweils 6–12 Interviews: 30 PT und mehr
Beteiligte	Analyst übernimmt die Führung, Entwickler und Produktmanager arbeiten aktiv mit.
Planung	Bei Projektinitialisierung; in den ersten Phasen des Projekts

Modellierte Realität: Personas und Szenarien

Eine Versicherungsgesellschaft entwickelt ein Softwaresystem, um Schadensmeldungen und die daraufhin geleisteten Entschädigungen zu verwalten. In der Spezifikation erscheint der Akteur „Sachbearbeiter“ und als einer der wesentlichen Anwendungsfälle „Schadensmeldung erfassen“. In einem Workshop, in dem ein erster Prototyp mit einigen Benutzern besprochen wird, verläuft die Diskussion etwas hitzig: Während ein Sachbearbeiter den Vorschlag ganz gut findet, da alle notwendigen Informationen übersichtlich dargestellt seien, stöhnt ein anderer über die vielen Eingabefelder und Abhängigkeiten. Es stellt sich heraus, dass der Erste das bestehende System täglich benutzt und viele Fälle erfasst, während der andere nur einige Male pro Monat Daten für wenige Spezialfälle eingibt und deshalb viele der Funktionen gar nicht benötigt. Der Prototyp erfüllt offenbar nur die Anforderungen des ersten Benutzers, nicht aber jene des zweiten.

Dieser Abschnitt befasst sich mit der Methode Personas und Szenarien. Es handelt sich dabei um zwei Instrumente, um die unterschiedlichen Bedürfnisse der Benutzer zu modellieren und daraus passende Lösungen abzuleiten.

Personas

Personas stellen prototypische Benutzer dar und verkörpern ihre unterschiedlichen Ziele, Verhaltensweisen und Eigenschaften, die im Hinblick auf das zu entwickelnde Produkt relevant sind. Die Methodik wurde vom Interaktionsdesigner Alan Cooper eingeführt und publik gemacht [Cooper & Reimann 07]. Die Namensgebung leitet sich vom

griechischen Theater der Antike ab. Die *Persona* war eine Maske, welche die Rolle der Schauspieler typisierte und gleichzeitig als Schallverstärker diente – ein treffender Begriff, wie wir finden. Die folgende Abbildung zeigt ein Beispiel einer Persona:



Jakob

- Sachbearbeiter Schadensabteilung
- 43 Jahre alt
- kaufmännische Ausbildung
- seit 24 Jahren im Versicherungsgeschäft
- seit 7 Jahren in der Schadensabteilung
- arbeitet täglich mit dem heutigen System

„Ich will ein System, wo ich alles auf einen Blick sehe.“

„Ich brauche kein intelligentes Programm. Ich weiß genau, was wohin gehört.“

Personas werden aufgrund von Informationen über die zukünftigen Benutzer eines Systems erarbeitet. Dazu dienen beispielsweise Ergebnisse aus Workshops mit Benutzern, Contextual Inquiries, Fragebögen oder Usability Walkthroughs mit bestehenden Systemen. Der Analyst entwirft Vorschläge und validiert diese mit den Beteiligten, oder die Personas werden in einem gemeinsamen Workshop erstellt. Eine Persona sollte schließlich die für das Produktdesign relevanten Eigenschaften der Benutzer widerspiegeln.

Im obigen Beispiel könnten zwei Personas, nennen wir sie Jakob und Niklaus, für zwei unterschiedliche Benutzergruppen stehen. Jakob und Niklaus unterscheiden sich deutlich in ihren Zielen und arbeiten ganz unterschiedlich mit der Applikation. Während Jakob mehrere Fälle pro Tag bearbeitet und das heutige System sehr gut kennt, arbeitet Niklaus nur gelegentlich am System für einzelne, komplexere Fälle.

Was Sie über Ihre Benutzer festhalten sollten

Im Rahmen eines Projekts entstehen mehrere Personas, die jeweils einen typisierten Benutzer beschreiben. Eine Persona kann über folgende Eigenschaften Auskunft geben:

- Ziele der Benutzer
- Beruf, Funktion, Verantwortlichkeiten und Aufgaben
- Fachliche Ausbildung, Wissen und Fähigkeiten

Die 7±2 wichtigsten Usability-Methoden

- Verhaltensmuster und Vorgehensweisen
- Werte, Ängste, Sehnsüchte, Vorlieben
- Allgemeine Computerkenntnisse
- Kenntnisse über verwandte Produkte, Vorgängersysteme, Konkurrenzprodukte
- Verbesserungspotenzial in der heutigen Situation
- Erwartungen an eine neue Lösung

Der erstellte Charakter soll einprägsam sein. Seine Eigenschaften sollen einfach verinnerlicht werden können. Dazu kann eine Persona mit zusätzlichen passenden Informationen zum Leben erweckt werden:

- Name, Alter, Geschlecht
- Markige Charakterzüge
- Bild, Skizze, Porträt
- Passende Zitate aus Interviews
- Ein Tag im Leben von ...

Insbesondere die weichen Kriterien wie Ziele, Werte und Ängste erscheinen einem Ingenieur im ersten Moment zwecklos. Die Handlungsweise von Menschen wird indessen stark von genau solchen Faktoren beeinflusst.

Fokussierung der neuen Lösung

Das Projektteam lässt den Blick über die Galerie der Charaktere schweifen. Ein Analyst meint: „Jakob und Niklaus sind sehr verschieden. Müssen wir da nicht zwei User Interfaces erstellen?“ Die Diskussion ist eröffnet. Wie soll das Projektteam mit den unterschiedlichen Personas umgehen? Gibt es wichtigere und weniger wichtige Benutzergruppen?

Personas lassen sich folgendermaßen klassifizieren:

Typ	Bedeutung
Primäre Persona	Für deren Bedürfnisse und Anforderungen wird das Produkt optimiert und die Benutzerschnittstelle erstellt.
Sekundäre Persona	Bedürfnisse sind größtenteils durch eine primäre Persona abgedeckt. Nur kleine Erweiterungen notwendig.
Ergänzende Persona	Bedürfnisse sind vollständig durch eine primäre Persona abgedeckt.
Non-Persona	Eine Persona, die vom Projektteam explizit nicht berücksichtigt wird.

Die Einstufung der Personas Jakob und Niklaus ist eine bewusste Entscheidung. Handelt es sich um zwei primäre Personas, dann wird das

Projektteam zwei optimierte Benutzerschnittstellen entwerfen. Wäre Niklaus hingegen eine Non-Persona, dann würde bewusst keine Optimierung für diese Benutzergruppe erfolgen.

Denkanstoß: *Denken Sie an ein spannendes Projekt zurück. Wurde über Benutzer diskutiert? Wurden gewisse Benutzerkreise ausgeklammert, bewusst oder unbewusst? Wie hätte die Verwendung von Personas dieses Projekt beeinflusst?*

Szenarien

Anwendungsszenarien oder kurz *Szenarien* sind ein zentrales Element in jeder benutzerorientierten Entwicklung. Sie schlagen die Brücke zwischen den Anforderungen und dem Entwurf einer neuen Lösung.

Ein Szenario beschreibt in Form eines realistischen Beispiels, wie ein Benutzer mit dem geplanten System interagieren wird. In einfachen Sätzen oder mittels Aufzählungspunkten wird ein konkreter Ablauf aus Benutzersicht im Anwendungskontext dargestellt. Dabei sollte, wie auch bei Personas, mehr auf inhaltlich richtige Aussagen als auf deren formale Korrektheit geachtet werden. Das folgende Beispiel zeigt ein kurzes Szenario zur Beschreibung einer neuen Versicherungsapplikation:

Szenario 1: Aufnehmen eines Schadensfalls

Es ist 15:00 Uhr. Bei Jakob klingelt das Telefon. Auf seinem Bildschirm erscheinen neben der Telefonnummer Name und weitere Angaben des anrufenden Kunden. Jakob nimmt den Anruf entgegen und begrüßt den ungeduldigen Kunden, der eine kaputte Fensterscheibe melden möchte. Da für diesen Kunden verschiedene Versicherungsverträge bestehen, wählt Jakob die entsprechende Police aus der Übersicht aus. Danach nimmt Jakob den Schadensfall des Kunden auf.

Das obige Szenario beschreibt in wenigen Sätzen, wie die Aufnahme eines Schadensfalls in Zukunft mit dem neuen System ablaufen soll. Es spiegelt eine Reihe von zusammengehörigen Anforderungen aus Benutzersicht wider:

- Automatische Anzeige des Namens und Angaben des Anrufers
- Darstellung aller bestehenden Versicherungspolicen eines Kunden
- Kurze Antwortzeiten des Systems

Szenarien werden basierend auf den Anforderungen an ein neues System erstellt. Sie können iterativ entwickelt oder in Workshops zusammen mit Benutzern erarbeitet werden. Ein großer Vorteil von Szenarien ist ihre leichte Verständlichkeit. Sie können von verschiedenen Stellen wie Auftraggeber, Benutzer und Entwicklung schon zu einem frühen Zeitpunkt überprüft, ergänzt oder korrigiert werden. Mit anderen Worten: Der Analyst *modelliert* mit Szenarien die Anforderungen an ein neues System. Folgende Eigenschaften zeichnen ein Szenario aus:

- Es wird für eine bestimmte Benutzergruppe entworfen, berücksichtigt ihre Eigenschaften und erfüllt ihre Bedürfnisse.
- Es stellt einen konkreten Fall aus der Anwendung dar.
- Es zeigt, wie die Benutzer die neue Software in ihrem realen Umfeld einsetzen werden.
- Es illustriert die für die Entwicklung der neuen Lösung relevanten Aspekte.
- Es beschränkt sich nicht auf den Schönwetterfall, sondern beschreibt auch exemplarisch wichtige Ausnahme- und Fehlersituationen.

Verwendung von Szenarien

Szenarien können zu unterschiedlichen Zeitpunkten in der Entwicklung einer neuen Lösung und für verschiedene Ziele eingesetzt werden:

- *Erhebung und Validierung von Anforderungen*: Die Reflektion am konkreten Beispiel erlaubt es Auftraggebern und Benutzern, Anforderungen in der konkreten Anwendungssituation zu vergegenwärtigen, zu überprüfen und zu ergänzen. Szenarien können als erste Prototypen eines neuen Systems betrachtet werden.
- *Spezifikation*: Szenarien illustrieren die Anwendung im realen Kontext und dienen als Ergänzung des Use-Case-Modells (siehe Abschnitt „Use Cases“, S. 48). Sie vermitteln den Entwicklern ein Verständnis der Abläufe und Zusammenhänge.
- *User-Interface-Konzept*: Design-Szenarien dienen dazu, die Abläufe der Benutzerschnittstelle zu beschreiben. Damit kann die Interaktion modelliert und mit Benutzern optimiert werden. Die technischen Anforderungen können von Entwicklern überprüft werden.
- *Usability-Testszenarien* (siehe Abschnitt „Usability Testing“, S. 60): Szenarien dienen als Basis für die Evaluation eines Systems oder eines Prototypen zusammen mit Benutzern.

- *Testszenarien*: Die erstellten Szenarien sind nützliche Instrumente, um daraus Testszenarien für das Software Testing im Rahmen der Entwicklung des Systems abzuleiten.
- *Schulung*: Szenarien dienen zur Schulung von Benutzern und als Basis für die Erstellung von Anleitungen.

Diese Durchgängigkeit über den gesamten Entwicklungsprozess macht Szenarien zu einem äußerst effektiven Instrument in der Entwicklung von interaktiven Systemen. Für eine weiterführende Vertiefung der szenariobasierten Entwicklung möchten wir an dieser Stelle auf das Buch [Rosson & Carroll 02] verweisen.

Hintergrund: Die Macht des guten Beispiels

Ein Analyst achtet darauf, formal korrekt und präzise zu formulieren. Die Spezifikation eines neuen Systems darf schlussendlich nur wenig Interpretationsspielraum zulassen. Um formal korrekt zu formulieren, muss zwangsläufig über verschiedene mögliche Fälle generalisiert werden. Die Gefahr dabei ist, dass die Realität auf der Strecke bleibt.

Der Sachbearbeiter nimmt eine Schadensmeldung eines Kunden auf ist eine formal korrekte Formulierung des Sachverhalts im obigen Beispiel, sie sagt allerdings wenig über die tatsächliche Situation aus. Natürlich wird im Beispiel nicht nur Jakob mit dem neuen System arbeiten. Es sollen damit auch nicht nur Schadensmeldungen für kaputte Fensterscheiben erfasst werden. Eine für die Usability des Systems zentrale Anforderung ist im vorliegenden Fall, dass der Benutzer aufgrund der Kundenanfrage (eine kaputte Fensterscheibe) schnell und eindeutig (denn der Kunde wartet am Telefon) die richtige Versicherungspolice (z. B. eine Hausratversicherung) zuordnen kann. Diese Anforderung wird erst durch die Schilderung der konkreten Anwendungssituation ersichtlich und kommt in einer formal korrekten, generalisierten Darstellung nur schwer zum Ausdruck.

Ein Beispiel dagegen ist weder eindeutig noch vollständig. Interessanterweise ist das menschliche Gehirn allerdings hervorragend dafür geeignet, aus Beispielen Regeln abzuleiten. Mittels weniger, guter Beispiele kann ein Sachverhalt oft schneller, umfassender und manchmal sogar präziser dargestellt werden als mit einer formalen Spezifikation. Personas und Szenarien nutzen diese Tatsache aus. Indem sie wichtige, stimmige und realistische Beispiele wiedergeben, können sie die Anwendung eines geplanten Systems schon früh im Entwicklungsprozess relativ genau umreißen, ohne dass die Details präzisiert werden müssen.

Die Benutzerperspektive

Mit Personas und Szenarien kann das Projektteam die Perspektive der Benutzer einnehmen und aus deren Sicht diskutieren. In erster Linie soll

damit das System oder Produkt entworfen und die Benutzerschnittstelle optimiert werden. Zum Beispiel könnte ein Szenario zeigen, in welcher Reihenfolge Jakob Informationen sucht, liest und eingibt. Der Vergleich zu einem analogen Szenario mit Niklaus würde die Unterschiede im Vorgehen sichtbar machen. So kann für unterschiedliche Benutzergruppen die richtige Mischung zwischen unterstützenden, beschränkenden und flexiblen Funktionen definiert werden.

Personas und Szenarien dienen einem Projektteam auch zur Beurteilung von Konkurrenz- oder Vorgängerprodukten: Wie gut wird eine Person den im Szenario skizzierten Fall lösen? Dies gibt wertvolle Hinweise über Stärken und Schwächen anderer Lösungen.

Durch die Benutzerperspektive ändert sich die Diskussion. Die Teilnehmer werden zur *Perspektivenübernahme* (siehe Kapitel „Einführung“, S. 1) ermuntert. Sie diskutieren aus der Sicht der Persona, statt aus einer aufgrund eigener Erfahrungen oder klischeehaften Vorstellungen gebildeten Individualsicht. Die Diskussion wird dadurch objektiver. Statt darüber zu streiten, ob der Benutzer dieses oder jenes Konzept versteht oder nicht, lässt sich die Frage untersuchen, welche Persona das Konzept kennt. Je fundierter die Daten, die zu den Personas führten, desto objektiver die Diskussion.

Darauf sollten Sie achten

- Wenn immer möglich sollten Personas aufgrund von Erkenntnissen über die (zukünftigen) Benutzer abgeleitet werden, beispielsweise aus Ergebnissen von Interviews, Contextual Inquiries, Beobachtungen oder Befragungen. Es besteht sonst die Gefahr, dass aufgrund falscher Vorstellungen an der Zielgruppe vorbeientwickelt wird.
- Es ist nicht immer möglich, mit Benutzern zu sprechen. Hier leisten Personas einen wertvollen Dienst, um Annahmen über die Benutzer aufzudecken, unterschiedliche Verständnisse zu diskutieren und ein gemeinsames Verständnis zu erreichen. In diesem Fall können die Beiträge von Fachleuten, Callcentern, Schulungen und andere sekundäre Informationsquellen ausgewertet werden.
- Personas erlauben einem Projektteam, bewusst auf die relevanten Eigenschaften der Benutzer zu fokussieren. Sie sind somit Teil der Projektabgrenzung und ein wichtiges Mittel für die Planung von Usability-Aktivitäten.
- Das Projektteam sollte die Anzahl Personas klein halten. Für jede primäre Persona wird eine eigene Benutzerschnittstelle bzw. eine eigene Sicht auf diese erstellt. Mehrere sekundäre Personas weisen

zudem darauf hin, dass die Ziele der Benutzerschnittstelle zu breit gefasst oder unklar sind.

- Bei Änderungen der Anforderungen im Verlauf des Projekts kann es relativ viel Aufwand bedeuten, Personas und Szenarien nachzupflegen und konsistent zu halten. Personas und Szenarien sind deshalb nur begrenzt als Modellierungswerkzeuge für Detailanforderungen geeignet. Ihre Stärke liegt in der Vermittlung von wichtigen Informationen mit Übersichtscharakter, zum Beispiel für eine Produktvision oder zur Ergänzung der Use Cases.
- Personas sind keine quantitativen Zielgruppenbeschreibungen. Personas verkörpern die für die Entwicklung relevanten Aspekte der Benutzer, während sich Zielgruppen auf die für Marketing und Verkauf wesentlichen Aspekte konzentrieren.
- Personas sind keine Marktsegmente. Segmente teilen die potenziellen Käufer entsprechend ihrer Eigenschaften in verschiedene Bereiche ein (z. B. alle Kunden im Alter von 18–35). Die Eigenschaften von Personas dagegen weisen keine Bereiche auf und erfüllen einen anderen Zweck. Sie spiegeln Bedürfnisse einer Benutzergruppe wider und fokussieren auf die Interaktion mit dem zukünftigen Produkt.

In Kürze

Methode	Personas und Szenarien
Resultate	Benutzergruppen im Detail charakterisiert Anwendungsszenarien ausgearbeitet Das Team nimmt die Benutzerperspektive ein.
Vorgehen	Für ein neues System die wesentlichen Eigenschaften der Benutzer zusammentragen, daraus Personas entwickeln und diese zum Leben erwecken. Szenarien erarbeiten, wie Benutzer mit dem neuen System umgehen. Personas einsetzen, um aus der Perspektive der Benutzer zu diskutieren und zu bewerten.
Aufwand	Modellieren von Personas: 2–6 PT Erarbeiten von Szenarien: 3–10 PT Inkl. Aufwand für Auftraggeber und Beteiligte. Diese Angabe geht davon aus, dass alle notwendigen Grundlagen vorhanden sind.
Beteiligte	Analyst erstellt Personas und Szenarien. Auftraggeber, Analyst, Software-Architekt, Entwickler, Fachleute usw. diskutieren aus der Sicht der Benutzer.
Planung	Personas entstehen bereits während der Projektabgrenzung und werden später verfeinert. Szenarien entstehen vor allem in der Detaillierungsphase, um Anforderungen und wesentliche Aspekte der Benutzerschnittstelle zu erarbeiten.

Einfach kommunizieren: Storyboards

In einer Fünftelsekunde kann man eine Botschaft rund um die Welt senden. Aber es kann Jahre dauern, bis sie von der Außenseite eines Menschenschädels nach innen dringt.

(Charles F. Kettering)

Software-Entwickler, Auftraggeber, Fachvertreter und Benutzer sprechen unterschiedliche Sprachen. Ein Beispiel stammt aus einem Projekt, in welchem der Software-Architekt von den Fachvertretern wissen wollte, ob ‚optimistic‘ oder ‚pessimistic locking‘ gefordert sei (keine Sorge, die beiden Begriffe sind hier nicht relevant). Der Software-Architekt stand vor der Entscheidung, auf welche Weise die neue Software damit umgehen sollte, wenn mehrere Benutzer zur gleichen Zeit die gleichen Daten bearbeiten wollen. Die Schwierigkeit bestand darin, dass er diese Frage nicht so formulieren konnte, dass die Fachvertreter diese auch verstehen und beantworten konnten.

Dieser Abschnitt stellt **Storyboards** vor, ein Mittel zur Kommunikation zwischen Auftraggebern, Benutzern und Entwicklern. Storyboards werden auch in anderen Gebieten, zum Beispiel in der Filmbranche, eingesetzt. Das Storyboard hilft dem Regisseur dabei, den Schauspielern und dem Filmteam den Aufbau des Films zu vermitteln. Es visualisiert Aspekte wie Perspektive, Beleuchtung, Gesichtsausdrücke, Kostüme und so weiter.

Die Anwendung visualisieren

Ein Storyboard zeigt mithilfe der Benutzerschnittstelle, wie ein System oder Produkt verwendet wird. Es stellt wichtige Aspekte der Anwendung bildlich dar und dient damit der Kommunikation zwischen allen Beteiligten. Im Wesentlichen handelt es sich dabei um die Visualisierung eines Szenarios (siehe Abschnitt „Szenarien“, S. 31).

Abhängig vom Kommunikationszweck kann ein Storyboard in unterschiedlichen Ausprägungen erstellt werden. Die Palette reicht von skizzenartigen oder realistisch gestalteten Abfolgen der Benutzerschnittstelle (*User Interface Storyboard*) bis zu Bildergeschichten, die auch Kontext und handelnde Personen darstellen.

Die folgende Abbildung zeigt einen Ausschnitt aus einem Storyboard:



Der Analyst setzt Storyboards in Situationen ein, wo Text alleine nicht ausreicht. Zwei wichtige Gründe sprechen für eine solche Visualisierung:

- In Bildern können Aspekte vermittelt werden, die mit Text nicht oder nur schwer auszudrücken sind, beispielsweise neuartige Konzepte, für die es noch keine Begriffe gibt.
- Mit der visuellen Umsetzung können Erlebnisse, die für die Anwendung von Bedeutung sind, besser in die Welt des Zielpublikums transportiert werden.

Ein Storyboard eignet sich deshalb, um folgende Gesichtspunkte aufzuzeigen:

- Dialogabläufe der Benutzerschnittstelle
- Schwer verständliche Konzepte oder Sachverhalte
- Wichtige Aspekte des Anwendungskontexts
- Spezielle oder komplexe Umgebungen, in denen das System eingesetzt wird

Eine Geschichte erzählen

Ein Storyboard erzählt die Geschichte, wie die Benutzer ein neues System nutzbringend einsetzen. Eine solche Geschichte vermittelt Vorschläge und Entscheidungen über Funktionsumfang, Gestaltung, Software-Architektur und mehr. Das Storyboard stellt eine implizite Frage an die Zuhörer: „Wir als Projektteam denken, dass diese Lösung eure Bedürfnisse erfüllt und so realisiert werden kann. Wo irren wir uns und wo habt ihr Bedenken?“ Damit dies erreicht werden kann, sollten die folgenden Aspekte beachtet werden:

- Die Geschichte erzählt ein konkretes Fallbeispiel.
- Sie ist örtlich und zeitlich eingeordnet.

Die 7±2 wichtigsten Usability-Methoden

- Sie erklärt die Zusammenhänge und stellt die kritischen Punkte detailliert dar.
 - Der dargestellte Fall sollte kein Trivialfall sein.
 - Die handelnden Personen werden charakterisiert.
 - Die Geschichte begründet plausibel, warum die Personen so handeln.
- Die Realitätsnähe und die Details der kritischen Punkte bieten Anlass zu interessanten Diskussionen, in denen Missverständnisse und Diskrepanzen aufgedeckt werden.

Was sollte ein Storyboard enthalten?

Ein Storyboard wird mit den Erkenntnissen im Verlauf eines Projekts präzisiert. Sind am Anfang nur erste Ideen oder verschiedene Varianten skizziert, so beinhaltet ein Storyboard später die bereits getroffenen Entscheidungen. Es enthält Aussagen zu folgenden Aspekten des geplanten Systems:

- Berücksichtigte und nicht berücksichtigte Bedürfnisse
- Änderungen der Geschäftsprozesse
- Neuerungen in der Arbeitsweise
- Enthaltene bzw. ausgeklammerte Funktionalität
- Den grundsätzlichen Aufbau der Benutzerschnittstelle
- Ausgewählte User-Interface-Details

Diese Liste ist nicht vollständig. Abhängig vom Kommunikationszweck müssen auch nicht zu jedem Punkt Aussagen vorhanden sein.

Zielgerichtet kommunizieren

Storyboards können in verschiedenen Situationen und für unterschiedliche Zwecke eingesetzt werden:

- Zur Diskussion einer Idee oder einer ausgearbeiteten Lösung mit Benutzern und weiteren Stakeholdern.
- Um das korrekte Verständnis der Bedürfnisse und der fachlichen Zusammenhänge zu prüfen und Missverständnisse auszuräumen.
- Zur Diskussion von Vor- und Nachteilen verschiedener Varianten.
- Um über Neuerungen zu informieren und damit beispielsweise Akzeptanz für das neue Werkzeug zu erzeugen.
- Um neugierig auf das Neue zu machen.
- Um Führungskräfte darüber zu informieren, wie ihre Vision durch die neue Lösung Realität wird.
- Um Entwicklern die relevanten Anforderungen der Benutzung näherzubringen und zu zeigen, warum gewisse Entscheidungen getroffen wurden.

- Um Benutzern im Rahmen einer Ausbildung einen Überblick über das System zu geben.
- Für Projektmarketing bei Auftraggebern, Geschäftsleitung und Benutzern.

Die Realität in den Workshop holen

Storyboards sind ein ausgezeichnetes Mittel, um die Realität der Anwendung in einen Workshop einzubringen. Die Teilnehmer diskutieren basierend auf der erzählten Geschichte über Annahmen, reflektieren Unterschiede zur heutigen Situation und klären Missverständnisse an realen Beispielen.

Möchte der Software-Architekt in unserem obigen Beispiel also wissen, wie die Applikation reagieren soll, wenn mehrere Benutzer gleichzeitig an den Daten arbeiten, dann kann ein User Interface Storyboard als Grundlage für eine angeregte Diskussion mit den Fachvertretern dienen. Diese können so die Konsequenzen für ihre Arbeit abschätzen und die beste Lösung wählen. Der Software-Architekt wird in einem solchen Workshop nicht nur eine fundierte Antwort auf seine Frage erhalten, sondern gleichzeitig viel über die Benutzer und ihre Arbeit lernen.

In Kürze

Methode	Storyboards
Resultate	Anwendung des neuen Systems aufgezeigt Akzeptanz bei Auftraggebern und Benutzern erzeugt Feedback zu Ideen und Entscheiden erhalten Kontext an das Projektteam kommuniziert
Vorgehen	Abläufe aus Benutzersicht aus vorhandenen Informationen zusammenstellen und visualisieren. Mit Auftraggebern, Benutzern und Mitgliedern des Projektteams validieren und Änderungen einarbeiten.
Aufwand	Pro Storyboard ca. 1–2 PT. Diese Angabe geht davon aus, dass alle notwendigen Grundlagen vorhanden sind.
Beteiligte	Analyst erstellt das Storyboard. Auftraggeber, Benutzer, Software-Architekt, Entwickler geben Feedback.
Planung	Früh im Projekt für Projektmarketing bei Sponsoren Im Requirements Engineering, um Feedback einzuholen In späteren Phasen zur Einführung und Ausbildung

Kritzeln für Fortgeschrittene: UI Prototyping

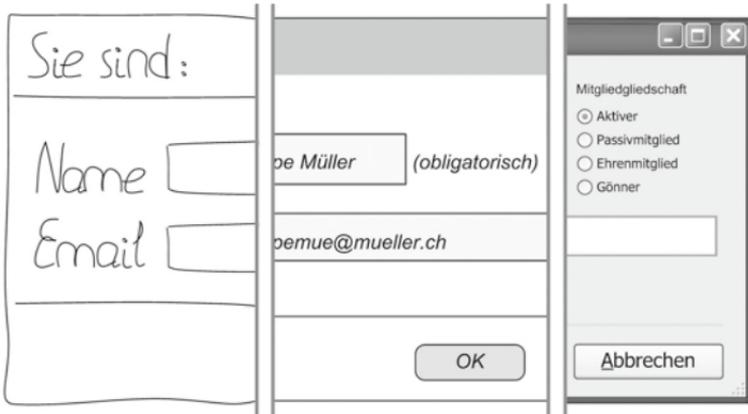
Ingenieure erstellen Prototypen, um ausgewählte Merkmale einer neuen Lösung, zum Beispiel deren Funktionsfähigkeit oder die eingesetzte Technologie, unter Beweis zu stellen. User Interface Prototyping (**UI Prototyping**) wird im Usability Engineering eingesetzt, um Aspekte der Benutzerschnittstelle zu entwerfen, zu evaluieren und zu verbessern, noch bevor ein lauffähiges System vorhanden ist. Weil dabei oft einfachste Werkzeuge wie beispielsweise Papier und Bleistift zum Einsatz kommen, spricht man auch von **LoFi Prototyping** (von englisch *Low Fidelity*: geringe Wiedergabetreue).

Dimensionen eines UI-Prototypen

Abhängig vom Ziel, das verfolgt wird, können unterschiedliche Arten von UI-Prototypen zum Einsatz kommen. Um den geplanten Prototypen näher zu charakterisieren, lassen sich die folgenden Dimensionen unterscheiden:

- *Funktionsumfang*: Wie viel der vorgesehenen Funktionalität der Benutzerschnittstelle soll im Prototyp gezeigt werden? Sind dies ausgewählte Ausschnitte, oder geht es darum, den gesamten Umfang darzustellen?
- *Funktionsiefe*: Wie detailliert sollen die einzelnen funktionalen Elemente wiedergegeben werden? Sollen beispielsweise mehrstufige Berechnungen nur angedeutet werden, oder sind die Zwischenschritte und ihre Resultate entscheidend?
- *Darstellungstreue*: Wie ähnlich soll der Prototyp dem Endprodukt in Bezug auf Aussehen der Benutzeroberfläche (Look&Feel) sein?
- *Interaktivität*: Wie interaktiv soll der Prototyp sein? Braucht es lauffähige Beispiele, um komplexe Abläufe wiederzugeben, oder genügen statische Darstellungen der Benutzerschnittstelle?
- *Datengehalt*: Sollen reale Daten zum Einsatz kommen, genügen realistische Beispiele oder gar Platzhalter für Bezeichnungen und dargestellte Informationen? Wie relevant ist die dargestellte Menge an Informationen?
- *Technische Reife*: Wie viel der endgültigen User-Interface-Technologie soll im Prototypen verwendet werden? Muss der Prototyp mit der Entwicklungsumgebung der Zielplattform entwickelt werden, oder sind einfache Zeichnungswerkzeuge ausreichend?

Die folgende Darstellung zeigt unterschiedliche Ausprägungen eines UI-Prototypen bezüglich Darstellungstreue: Einfache Papierskizze, Drahtmodell (*Wireframe*) und endgültiges Look&Feel.



Jeder Prototyp stellt einen Kompromiss zwischen notwendigem Aufwand und Zweck dar. Bevor Sie mit UI Prototyping loslegen, sollten Sie sich deshalb im Klaren sein, welche Fragestellungen Sie verfolgen. Daraus lässt sich ableiten, welche Art von UI-Prototyp geeignet ist. Die folgenden Abschnitte beinhalten einige typische Verwendungszwecke für UI-Prototypen.

Anforderungen klären

Ein Analyst skizziert aufgrund der mittels Contextual Inquiry vor Ort gesammelten Informationen erste Entwürfe der Benutzerschnittstelle mit Papier und Bleistift. Bestehende Formulare und Applikationen geben Auskunft über die benötigten Informationen. Die beobachteten Abläufe zeigen, in welcher Reihenfolge die Benutzer diese Informationen verwenden. Die Entwürfe werden zu einer ersten Simulation der Benutzerschnittstelle zusammengestellt und mit den Benutzern besprochen.

Man spricht bei Attrappen der Benutzerschnittstelle wie im obigen Beispiel auch von **Mock-ups**. Mit Mock-ups können Benutzer bereits konkrete Fälle durchspielen und diskutieren. Dabei geht es nicht darum, das System zu entwerfen, sondern die Anforderungen der Benutzer mit einem einfachen Hilfsmittel erfahrbar zu machen. Der Analyst

Die 7±2 wichtigsten Usability-Methoden

deckt dabei neben fachlichen Missverständnissen auch weitere Bedürfnisse der Benutzer auf:

- Notwendiger Informationsgehalt
- Funktionalität und Abläufe
- Einbettung in die Geschäftsprozesse
- Datenaustausch mit anderen Systemen und Applikationen
- Darstellung von Tabellen, Grafiken, Funktionen usw.
- Wichtige Details der Benutzerschnittstelle

Die Benutzerschnittstelle konzipieren

Eine der zentralen Aufgaben im Usability Engineering ist es, ein für die Aufgaben der Benutzer geeignetes **User-Interface-Konzept** zu erarbeiten. Dabei geht es darum, die Grundsätze der Benutzerschnittstelle festzulegen. Wie bewegen sich die Benutzer durch Menüs und Dialoge? Wie wird die Information strukturiert und dargestellt? Muss das System für spezielle Technologien, zum Beispiel für die Bedienung per Touchscreen, optimiert werden?

Als Ausgangslage dienen die erarbeiteten Personas und Szenarien und die vereinbarten Anforderungen. Einfache Prototypen helfen dem User Interface Designer, die Szenarien durchzuspielen und ein geeignetes Konzept auszuarbeiten. Das User-Interface-Konzept sollte schließlich folgende Aspekte beinhalten:

- Grundsätzlicher Aufbau und Screen-Layout
- Aufteilung und Struktur von Informationen
- Verwendung und Verhalten von Fenstern
- Wichtige Bedienelemente
- Navigation mittels Menüs, Schaltflächen und Links
- Prüfung von Eingaben und Anzeige von Fehlermeldungen
- Konzepte für das Speichern von Informationen und Zuständen
- Rückgängig machen und erneut ausführen
- Interaktionsprinzipien wie direkte Manipulation, Drag&Drop oder Kontextmenüs

Dabei muss auch die Technologie der Zielplattform berücksichtigt werden, beispielsweise Eingabe- und Ausgabemedien, Betriebssystem, Bildschirmgrößen und Auflösung.

Die Benutzerschnittstelle optimieren

Gerade wenn hohe Anforderungen an Effizienz, Verständlichkeit oder an die Qualität der Arbeitsergebnisse gestellt werden, gewinnen Details

der Benutzerschnittstelle große Bedeutung. Schon eine unglücklich formulierte Bezeichnung kann verhindern, dass Benutzer einen Automaten richtig benutzen können. Eine immer wieder zu bestätigende Warnmeldung wird Vielbenutzer zur Weißglut treiben, und umständliche Mausoperationen verlangsamen die Arbeit.

Ein User Interface Designer sollte deshalb kritische Ausschnitte der Benutzerschnittstelle mittels Prototypen umsetzen und mit Benutzern evaluieren (siehe Abschnitt „Usability Testing“, S. 60). Im Folgenden einige wesentliche Fragestellungen:

- Erlaubt die Benutzerschnittstelle flüssiges Arbeiten?
 - Gibt es Hürden oder Stolpersteine für Personen, die das System zum ersten Mal benutzen?
 - Ist die Navigation effizient?
 - Finden Benutzer die gewünschte Information?
 - Werden Warnmeldungen bemerkt und richtig interpretiert?
 - Passt die Benutzerschnittstelle zu den Details der Arbeitsabläufe?
- Kritische Funktionen sollten mit einer realistischen Menge echter Daten hinterlegt sein. Damit können Benutzer ausgewählte Fälle durchspielen und die Benutzerschnittstelle auf ihre Tauglichkeit beurteilen.

Für gutes Aussehen sorgen

Ein nicht unerheblicher Aspekt moderner Benutzeroberflächen ist ein ästhetisches Design. Grafikdesigner arbeiten mit Prototypen, um sich verschiedene Design-Varianten vor Augen zu führen und die Details der Gestaltung auszuarbeiten. Ein zentraler Punkt dabei ist, Funktionalität und Ästhetik zu verbinden. Der User Interface Designer wird sich deshalb zu folgenden Punkten Gedanken machen:

- Anordnung und Abstände der Bedienelemente
- Ausrichtung der Beschriftungen
- Gruppieren von Elementen
- Farben und Kontraste
- Verwendung von Schriften
- Stil und Darstellung von Icons
- Weitere grafische Elemente

Für solche Betrachtungen ist im Regelfall ein Werkzeug notwendig, welches eine grafische Gestaltung der Bedienelemente ermöglicht. Zahlreiche Grafikprogramme bieten die häufigsten Bedienelemente moderner Benutzeroberflächen als vordefinierte Schablonen an und erlauben gleichzeitig, schnell und detailliert zu gestalten.

User Interface spezifizieren

Mit einem User-Interface-Prototypen können viele Aspekte der Benutzerschnittstelle auf anschauliche Weise festgehalten werden. Im Rahmen der Spezifikation kann ein Prototyp für folgende Zwecke eingesetzt werden:

- Illustration des Funktionsumfangs
- Verdeutlichung der Funktionsweise
- Spezifizieren der User-Interface-Elemente
- Aufzeigen der Navigation und Interaktion
- Visualisierung der geplanten Lieferung
- Abschätzung des Realisierungsaufwands durch die Entwickler

In der Regel sind im Rahmen der User-Interface-Spezifikation die Darstellung des Funktionsumfangs und für ausgewählte Aspekte auch eine realistische Funktionstiefe gefordert. Eine gewisse Interaktivität ist ebenfalls hilfreich, da so verschiedene Zustände des User Interface visualisiert werden können. Ein solcher Prototyp wird deshalb in vielen Fällen mit einer Technologie erstellt, welche der Zieltechnologie nahekommt. Bei einfacheren Produkten genügen auch Grafik- und Bildbearbeitungsprogramme.

Die folgende Tabelle fasst die beschriebenen Verwendungszwecke und die dafür notwendigen Dimensionen für die Erstellung von Prototypen zusammen:

Zweck	Dimensionen
Anforderungen klären	Großer Funktionsumfang (in mehreren Prototypen) mit realistischen Daten
Benutzerschnittstelle konzipieren	Mittlere Darstellungstreue Ausgewählte Funktionen im Detail Teilweise interaktiv
Benutzerschnittstelle optimieren	Hohe Darstellungstreue Interaktiv für ausgewählte Funktionen Oft reale Daten notwendig Oft hohe technische Reife notwendig
Für gutes Aussehen sorgen	Hohe Darstellungstreue
User Interface spezifizieren	Funktionsumfang und -tiefe sind mittel bis hoch Mittlere bis hohe Interaktivität

Paper Prototyping

Leider sind wir Erwachsenen der Ansicht, dass Zeichnen nur etwas für Kinder oder Künstler ist, jedoch keine seriöse und ernsthafte

Tätigkeit. In diesem Abschnitt möchten wir in aller Deutlichkeit darstellen: Zeichnen ist eine Notwendigkeit! Der Entwurf eines ersten Prototypen mit Papier und Bleistift hat verschiedene Stärken:

- Praktisch alle Personen können damit umgehen.
- Einfache Skizzen sind schnell erstellt und angepasst.
- Es wird weniger Zeit für Details aufgewendet als mit Grafikprogrammen.
- Es sind keine technischen Hilfsmittel notwendig.
- Auch nicht standardisierte Bedienelemente sind schnell skizziert.
- Mehrere Personen können gemeinsam arbeiten.
- Einen Papier-Prototypen zu zerknüllen und wegzuwerfen, fällt leicht.

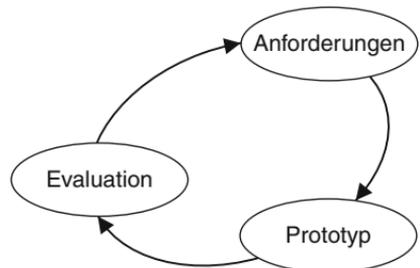
Aus diesen Gründen arbeitet beispielsweise ein Analyst bei der Bedürfnisanalyse effizienter mit Papier und Bleistift als mit elektronischen Werkzeugen. Papier-Prototypen sind auch gut geeignet, um im Rahmen von Interviews und Workshops aufkommende Ideen zu visualisieren.

Ein weiterer wichtiger Punkt ist, dass Mock-ups auf Papier und elektronische Prototypen nicht die gleiche Wirkung haben. Papier-Prototypen signalisieren durch ihre Skizzenhaftigkeit, dass noch viel offen ist und auch über Grundsätzliches diskutiert werden kann. Entsprechend lässt sich gezielter über Abläufe und den konzeptionellen Aufbau diskutieren. Bei einem endgültig aussehenden Prototypen gehen hinzugezogene Personen eher davon aus, dass das Grobkonzept bereits feststeht und nur noch an den Details gefeilt werden soll. Eine umfassende Übersicht zum Thema bietet das Buch *Paper Prototyping* [Snyder 03].

Iteratives Vorgehen

User Interface Prototyping ist ein iterativer Prozess. Auf das Wesentliche reduziert stellt dies die nebenstehende Abbildung dar:

Das Projektteam erstellt aufgrund der Anforderungen einen ersten UI-Prototypen des geplanten Systems. Dieser hilft bei der weiteren Optimierung und Präzisierung der Anforderungen, beispielsweise im Rahmen von Usability-Tests und Walkthroughs (mehr



Die 7±2 wichtigsten Usability-Methoden

dazu im Abschnitt „Usability Testing“, S. 60) sowie Workshops oder Reviews.

Einige gängige Prototyping Tools

User-Interface-Prototypen können mit verschiedenen Werkzeugen erstellt werden. Im Folgenden finden Sie einige Möglichkeiten:

Tool

Papier & Bleistift, Whiteboard, Folien

Office-Anwendungen

Microsoft Powerpoint, Apple Keynote

Bildbearbeitungsprogramme

Adobe Photoshop

Grafikprogramme

OmniGraffle, Microsoft Visio, Adobe Fireworks

UI-Prototyping-Werkzeuge

Axure RP, Balsamiq Mockups

Multimediawerkzeuge

Adobe Director, Adobe Flash

Programmierwerkzeuge

HTML-Editoren, Entwicklungswerkzeuge wie Microsoft Expression Blend und Adobe Flex

Verwendung

Diese Mittel eignen sich besonders in Workshops und Interviews.

Mit wenig Aufwand können damit erste Interaktionen erstellt werden.

Volle Kontrolle über die grafische Gestaltung.

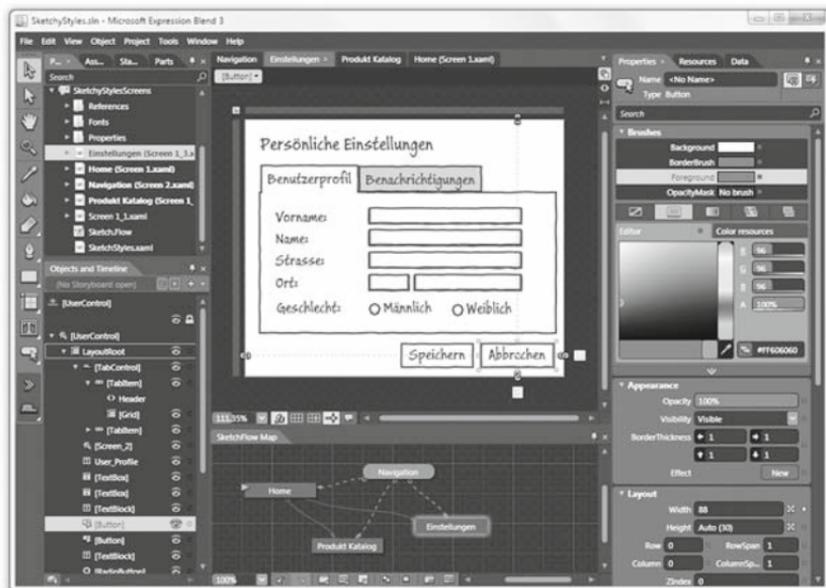
Bieten vordefinierte Schablonen der gängigen Bedienelemente. Damit lassen sich schnell gut und echt aussehende Mock-ups erstellen.

Speziell für UI Prototyping entwickelt, unterstützen diese Werkzeuge den Designprozess von der interaktiven Skizze bis zur Umsetzung auf der Zielplattform.

Insbesondere für interaktive Prototypen mit hohen Design-Anforderungen oder Animationen.

Interaktive Prototypen mit größeren Datenmengen und komplexerem Verhalten.

Die folgende Abbildung zeigt einen mit Microsoft Expression Blend SketchFlow® erstellten skizzenartigen, jedoch beliebig interaktiven User-Interface-Entwurf, der für das UI Prototyping verwendet werden kann.



Darauf sollten Sie achten

- User Interface Prototyping ist eine iterative Tätigkeit. Investieren Sie nicht zu viel Zeit in die Perfektionierung, bevor Sie Feedback einholen.
- Es lohnt sich, ein Werkzeug zu wählen, mit dem ohne großen Aufwand verschiedene Varianten ausgearbeitet und Änderungen umgesetzt werden können. Häufig wird zu früh auf Programmierwerkzeuge für die Erstellung von UI-Prototypen gesetzt.
- Bei Geräten sollten Prototypen die Software in Kombination mit der Hardware zeigen. Anzahl, Größe und Anordnung von Bedienelementen beeinflussen das Konzept auf dem Display maßgeblich.

In Kürze

Methode	User Interface Prototyping
Resultate	Anforderungen evaluiert Details der Benutzerschnittstelle erarbeitet Benutzerschnittstelle optimiert
Vorgehen	Erstellen von Prototypen und Evaluieren mit Benutzern

Die 7±2 wichtigsten Usability-Methoden

Aufwand	Der Aufwand hängt sehr stark vom Zweck des Prototypen ab. Einige Größenordnungen für die Erstellung: Papier: Minuten bis Stunden Visio: Stunden bis Tage Photoshop: Tage bis Wochen Entwicklungstools: Tage bis Monate
Beteiligte	Analyst, um erste Ideen zu evaluieren. User Interface Designer, um Funktionalität, dargestellte Informationen und das User-Interface-Konzept zu erarbeiten. Entwickler, Benutzer, Auftraggeber geben Feedback.
Planung	In den frühen Phasen eines Projekts, um Ideen zu konkretisieren und Anforderungen zu erheben. Während der eigentlichen Spezifikation, um die Details der Benutzerschnittstelle zu erarbeiten und zu evaluieren.

Für die Entwicklung dokumentieren: Use Cases

Anwendungsfälle (*Use Cases*) stammen aus dem Software Engineering und sind ein verbreitetes Instrument zur Spezifikation von technischen Systemen. Im RUP beispielsweise sind Use Cases ein zentrales Konzept.

In der Usability-Literatur findet man zum Thema dagegen eher wenig. Dies mag überraschen, da mittels Anwendungsfällen das Verhalten eines Systems aus Benutzersicht dargestellt wird. Die Abläufe, die ein Benutzer später mit dem System erlebt, werden zu einem großen Teil von den spezifizierten Anwendungsfällen bestimmt. Deren Entwurf hat für die Usability eines Systems daher eine zentrale Bedeutung. Auch wenn Use Cases und deren Modellierung keine Usability-Methode im engeren Sinn darstellen, haben wir uns aus oben genannten Gründen entschieden, sie in unsere Sammlung der wichtigsten Usability-Methoden aufzunehmen. Wir beschränken uns allerdings auf eine Übersicht und den Zusammenhang mit anderen Usability-Methoden.

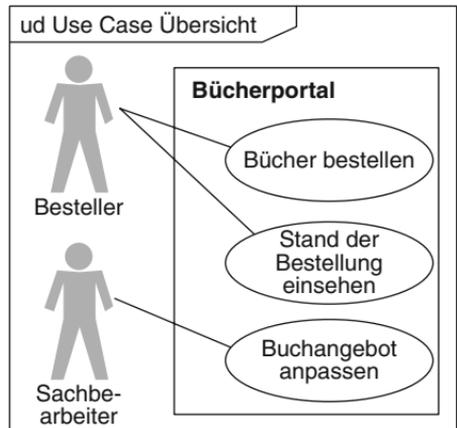
Das Use-Case-Modell

Use Cases beschreiben die (geplante) Funktionalität eines Systems und damit dessen Verhalten gegenüber der Außenwelt. Eine große Stärke von Use Cases ist, dass diese die Funktionsvielfalt aus Benutzersicht in zusammengehörige Einheiten aufbrechen und Schritt für Schritt näher definieren. Ein weiterer Vorzug von Use Cases ist die

Beschreibung in natürlichsprachlicher Form, die im Verlauf der Entwicklung für alle Beteiligten verständlich bleibt.

Um eine bestimmte Funktionalität darzustellen, verwendet der Analyst sogenannte **Akteure** (*Actors*), die mit dem System in Interaktion treten. Akteure verkörpern dabei die Rollen von Benutzern oder anderen Systemen. Der Anwendungsfall beschreibt den funktionalen Ablauf mit dem System aus Sicht des Akteurs. Soll beispielsweise die Funktionalität für eine Anwendung zur Buchbestellung im Internet beschrieben werden, dann wäre der Kunde, der das Buch bestellt, ein Akteur, die Bestellung eines Buches ein Anwendungsfall, die Verfolgung der Bestellung ein zweiter, die Bewertung eines Buches ein dritter usw. Dabei sollte ein Anwendungsfall immer eine aus Sicht des Akteurs abschließende Handlung umfassen.

Die Gesamtheit der Akteure und Anwendungsfälle eines Systems wird als *Use-Case-Modell* bezeichnet. Der Analyst modelliert damit im Rahmen der Anforderungsanalyse die Funktionsweise des Systems. Das System selbst wird dabei von außen als Blackbox betrachtet, d. h. es wird noch nicht beschrieben, wie das Verhalten zustande kommt. Die einzelnen Interaktionsschritte jedes Anwendungsfalls werden in einem zweiten Schritt genau beschrieben und dienen zur Spezifikation für die Entwicklung. Die Anwendungsfälle verkörpern somit das funktionale Verhalten eines Systems. Das Use-Case-Modell lässt sich auch grafisch darstellen. Ein solches *Use-Case-Diagramm* dient als Übersicht über die Funktionalität des Systems und seiner Schnittstellen zur Außenwelt. Die Abbildung zeigt ein Beispiel eines Use-Case-Diagramms in der UML-Notation.



Woher kommen die Akteure?

Akteure und Anwendungsfälle werden typischerweise in *Use Case Workshops* mit Auftraggebern, Fachstellen und Benutzern erarbeitet. Die Schwierigkeit dabei ist, dass Akteure keine realen Benutzer,

sondern Rollen darstellen, die mit dem System interagieren. Ein häufiger Fehler ist, dass Akteure so weit verallgemeinert werden, dass der Bezug zur Realität verloren geht. Dies kann dazu führen, dass wichtige Bedürfnisse vergessen oder Anforderungen von unterschiedlichen Benutzergruppen in einem einzigen Akteur vereinigt werden. In der Folge werden Anwendungsfälle konstruiert, die eine Vermischung von Funktionen beinhalten und besser klar getrennt würden; mit dem Resultat, dass die so spezifizierten Abläufe für die tatsächlichen Benutzer in der realen Anwendung später nicht geeignet sind.

Besser ist es, ein Use-Case-Modell aufgrund der Ergebnisse aus der Analyse von Benutzern und Kontext zu erstellen. Contextual Inquiry, Interviews oder vergleichbare Methoden, mit denen die Verantwortlichkeiten und Tätigkeiten der Benutzer aufgenommen werden, sind eine wichtige Voraussetzung dafür.

Die Use-Case-Spezifikation

In der Use-Case-Spezifikation wird ein Anwendungsfall detailliert beschrieben. Sämtliche Schritte in der Interaktion zwischen Akteur und System werden aufgeführt. Neben dem „Schönwetterfall“ berücksichtigt der Analyst hier auch alternative Abläufe und Fehlerfälle.

Bei der Ausformulierung eines Anwendungsfalls sollten noch keine technischen Details beschrieben werden. Ein häufiger Fehler ist, dass in Use-Case-Spezifikationen Details der Benutzeroberfläche vorweggenommen werden. Dies ist aus zweierlei Gründen nicht sinnvoll: Der beschriebene Ablauf kann unter Umständen mit einem alternativen Interaktionskonzept besser umgesetzt werden als zunächst angenommen. Weiter erschwert eine zu starke Detailtiefe die Pflege und Anpassung der Anwendungsfälle bei Änderungen. Es ist deshalb weitaus besser, neben der Use-Case-Spezifikation einen User-Interface-Prototypen oder ein Storyboard zu erstellen, das die Abläufe mit der konkreten Benutzerschnittstelle verdeutlicht. Mit einem solchen Prototypen kann der Analyst seinen Entwurf mit Auftraggebern und Benutzern verifizieren und verfeinern. Neben natürlichsprachlichen Beschreibungen in strukturierten Dokumenten werden ergänzend auch Ablaufdiagramme für die Darstellung der einzelnen Schritte verwendet.

Die Spezifikation muss schließlich von zwei ganz unterschiedlichen Parteien verstanden werden können: dem Auftraggeber und den

Entwicklern. Anwendungsfälle sollten deshalb sowohl formal korrekt als auch verständlich sein. Eine hervorragende Hilfestellung für die Formulierung guter und verständlicher Anwendungsfälle bietet [Cockburn 08].

Hintergrund: Funktionale und nicht-funktionale Anforderungen

Im Anforderungsmanagement wird zwischen funktionalen und nicht-funktionalen Anforderungen unterschieden:

Funktionale Anforderungen betreffen, wie der Name sagt, jene Aspekte, die mit der Funktionalität des geplanten Systems zusammenhängen. Ein Buchbestellsystem kann beispielsweise Funktionen zum Suchen, Bestellen und Bewerten eines Buches anbieten. Die funktionalen Abläufe, die für das Bestellen eines Buches notwendig sind, können etwa mit Use Cases näher beschrieben werden.

Nicht-funktionale Anforderungen umfassen sämtliche geforderten Rahmenbedingungen. Im Beispiel muss unser Buchbestellsystem bestimmten Anforderungen bezüglich Verfügbarkeit, Antwortzeiten, Ausfallsicherheit usw. genügen. Nicht-funktionale Anforderungen spielen ebenfalls eine wesentliche Rolle für die Benutzbarkeit eines Systems und haben Auswirkungen auf die verwendete Technologie und Software-Architektur. Auch bezüglich Usability können nicht-funktionale Aspekte festgelegt werden, etwa im Hinblick auf die geforderte Effizienz für die Benutzer. In unserem Beispiel: Die Suche und Bestellung eines bestimmten Buches soll für 90 % der Benutzer in weniger als fünf Minuten möglich sein.

Essential Use Cases

Die Überlegung, technische Details in Anwendungsfällen noch weitgehend auszuklammern, führte zum Konzept der **Essential Use Cases** [Constantine & Lockwood 99]. Dabei wird versucht, in der Beschreibung von Anwendungsfällen nur die Interaktion mit dem System als solche auszudrücken und von deren technischen Umsetzung konsequent zu trennen. Damit soll vermieden werden, dass zu früh auf eine – eventuell nicht optimale – technische Lösung eingeschwenkt wird. Das Konzept der Essential Use Cases fand in Usability-Fachkreisen großen Anklang. Das folgende Beispiel zeigt einen einfachen Essential Use Case für ein Bücherbestellsystem:

Die 7±2 wichtigsten Usability-Methoden

Essential Use Case: ein Buch für eine Bestellung vormerken

Absicht des Benutzers

führt Suche aus

[optional] informiert sich über Buch

fügt Buch zur Bestellung hinzu

Verantwortlichkeit des Systems

präsentiert Suchdialog

präsentiert passende Bücher

präsentiert Details zum Buch

bestätigt Abschluss

Use Cases oder Szenarien?

Akteure und Anwendungsfälle sind konzeptionell verwandt mit Personas und Szenarien (siehe S. 28). Beides sind Methoden, mit denen ein Analyst die Interaktionen mit dem System modelliert. Dennoch gibt es einige wesentliche Unterschiede:

- Akteure definieren Rollen, die mit dem System interagieren. Personas hingegen charakterisieren prototypische Benutzer und fokussieren auf die Eigenschaften von unterschiedlichen Benutzergruppen.
- Anwendungsfälle halten einen bestimmten Teil des Systemverhaltens für die Entwicklung fest. Sie generalisieren über die verschiedenen Möglichkeiten der Nutzung. Szenarien dagegen beschreiben konkrete Beispiele der Systemnutzung und illustrieren die Anwendung im realen Kontext.

Die folgende Tabelle verdeutlicht die unterschiedlichen Ziele:

Methode	Format	Ziel
Akteur	Rollenbeschreibung	Gruppierung zusammengehöriger funktionaler Abläufe
Persona	Prototypischer Benutzer	Charakterisierung unterschiedlicher Benutzergruppen
Anwendungsfall	Beschreibung des Systemverhaltens	Spezifikation des funktionalen Verhaltens für die Entwicklung
Szenario	Konkretes Beispiel der Systemnutzung	Beschreibung der Anwendung im realen Kontext

Die beiden Methoden ergänzen sich in der Praxis ausgezeichnet. Realistische Personas und Szenarien, die aufgrund von Benutzeranalysen erstellt wurden, bilden eine gute Grundlage für den Entwurf des Use-Case-Modells. Im Verlauf des Projekts detailliert und komplettiert der Analyst die Anwendungsfälle, sodass das gesamte funktionale Verhalten

ten des Systems, inklusive Anbindung von externen Systemen, spezifiziert wird.

Denkanstoß: Stellen Sie sich ein System zum Bestellen von Büchern vor. Denken Sie sich den Akteur „Besteller“ und den Use Case „Buch bestellen“.

Nun stellen Sie sich Ruth vor. Sie ist Mitarbeiterin einer wissenschaftlichen Bibliothek und ordert am Montag 219 Neuerscheinungen über dieses Bestellsystem.

Wie unterscheidet sich ein solches Bestellsystem von jenem, das Lara verwendet, um den neusten Roman ihres Lieblingsautors zu bestellen?

Ein Analyst, der eine Spezifikation für ein neues System schreibt, untersucht verschiedene einzelne Situationen und generalisiert die erhaltenen Informationen zu einem abstrakten Modell. Hat der Leser der Spezifikation allerdings nur dieses abstrakte Modell zur Verfügung, kann er die konkreten Einzelfälle und deren Unterschiede nicht herleiten. Denn diese Informationen sind in der Abstraktion nicht mehr enthalten. Beispiele hingegen sind hervorragende Instrumente, um Zusammenhänge oder Abläufe aufzuzeigen (siehe auch Abschnitt „Hintergrund: Die Macht des guten Beispiels“, S. 33). In den meisten Fällen ist es bereits hilfreich, wenige stellvertretende Beispiele – etwa einen typischen Arbeitsablauf eines typischen Benutzers – möglichst realitätsnah wiederzugeben.

Checkliste für den Einsatz von Use Cases

- Sind die erstellten Akteure in der Realität wiederzufinden? Wurden sie auf Basis der Analyse von Benutzern und Kontext erstellt? Wurden verschiedene Benutzergruppen berücksichtigt?
- Umfassen die Anwendungsfälle abgeschlossene Handlungsabläufe aus Benutzersicht? Entsprechen diese den erarbeiteten Szenarien und tatsächlichen Arbeitsprozessen der Benutzer?
- Wurde in den Use-Case-Spezifikationen an Alternativabläufe und das Verhalten bei Fehlerfällen gedacht? Auch diese gilt es in einem benutzerorientierten Systemdesign zu berücksichtigen.
- Wurden die Anwendungsfälle sowie alternativen Abläufe mit Angaben zu Dauer und Häufigkeit des Vorkommens ergänzt?
- Beinhalten die Beschreibungen keine technischen Details, die besser in einem User-Interface-Konzept, Storyboard oder Prototypen abgebildet würden?
- Ist bei jedem Schritt im Use Case klar, welche Informationen der Benutzer für seine Aufgabe benötigt?

Die 7±2 wichtigsten Usability-Methoden

In Kürze

Methode	Use Cases
Resultate	Übersicht über die Funktionalität eines Systems Detaillierte Beschreibung des Verhaltens des Systems aus Benutzersicht
Vorgehen	Ermitteln von Akteuren und Anwendungsfällen Entwurf eines Use-Case-Modells Spezifikation der Anwendungsfälle Review mit Stakeholdern
Aufwand	Entwurf und Review Use-Case-Modell: 3–10 PT Spezifikation stark abhängig vom funktionalen Umfang des Systems. 10–15 Use Cases: 15–30 PT
Beteiligte	Auftraggeber, Analyst
Planung	Im Rahmen der Anforderungsanalyse, im Anschluss an Contextual Inquiry oder bei Vorliegen von Personas und Szenarien

Usability Guidelines und Styleguides

*It's like a jungle sometimes
It makes me wonder how I keep from goin' under*
(Grandmaster Flash, 1982)

„Gibt es nicht einfach ein paar Richtlinien für gute Usability, damit unser Produkt besser benutzbar wird?“

Ja, ungefähr 600'000. Zumindest ist dies die Anzahl Treffer, wenn man im Internet nach „Usability Guidelines“ googelt. Halten Sie also einfach alle relevanten Regeln ein. Genauso wie ein Schriftsteller, der ein Buch schreibt oder ein Architekt, wenn er ein Haus baut. Alles klar?

Was sind Usability Guidelines?

Usability Guidelines beinhalten Regeln für die Gestaltung von Benutzeroberflächen. Darunter fällt zunächst ein sehr breites Spektrum von Regelwerken. Die Palette reicht von globalen Grundsätzen bis hin zu detaillierten Vorgaben. In der Praxis werden deshalb oft die folgenden Begriffe unterschieden:

- *(G)UI Guidelines*: eher generelle Richtlinien für die Verwendung und das Verhalten von (grafischen) User-Interface-Elementen.
- *Styleguides*: konkrete Vorgaben für die visuelle Gestaltung und das Layout einer bestimmten Benutzeroberfläche. Styleguides beschreiben Aussehen und Verhalten (Look&Feel) von User-Interface-Elementen, abhängig von der eingesetzten Technologie.

Verschiedene Arten von Guidelines

Usability Guidelines können bezüglich ihres Verwendungszwecks unterschieden werden. Die folgende Aufzählung erleichtert Ihnen die Einordnung, wenn Sie es mit Regelwerken im Bereich Usability zu tun bekommen:

Gesetzliche Verordnungen: Vorschriften, die in erster Linie den Gesundheitsschutz des Arbeitnehmers beim Umgang mit technischen (Bildschirm-) Geräten bezwecken. Es mag Sie vielleicht überraschen, dass Aspekte der Benutzerfreundlichkeit gesetzlich geregelt sind, eine nachweisbare Missachtung könnte für einen Arbeitgeber allerdings unerfreuliche Konsequenzen haben. Im EU-Raum diesbezüglich relevant ist die Richtlinie 90/270/EWG [EG 90], die unter anderem Vorschriften zu Mindestanforderungen an die Mensch-Maschine-Schnittstelle beinhaltet.

Normen: Nationale oder internationale Normen mit dem Ziel, durch Gestaltungsvorgaben die Anwendung von Technologien zu standardisieren und für die Benutzer zu vereinfachen. Prominentestes Beispiel ist die internationale ISO-Norm 9241, die in 17 Teilen „Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten“ definiert [DIN EN ISO 9241, 1997-2006]. Häufig verwendet werden die folgenden sieben Kriterien, die ein benutzerfreundliches Dialogsystem auszeichnen:

Grundsätze der Dialoggestaltung (DIN EN ISO 9241-10, 9241-110)

Aufgabengemessenheit	Das System unterstützt die Erledigung der Aufgaben und den Arbeitsablauf der Benutzer.
Selbstbeschreibungsfähigkeit	Das System enthält Erläuterungen und ist ausreichend verständlich.
Steuerbarkeit	Der Benutzer kann den Dialogablauf beeinflussen.
Erwartungskonformität	Erwartungen, Eigenschaften und Gewohnheiten der Benutzer werden unterstützt.
Fehlertoleranz	Fehler erfordern keinen oder nur geringen Korrekturaufwand.
Individualisierbarkeit	Das System kann an die individuellen Bedürfnisse angepasst werden.
Lernförderlichkeit	Das System erfordert einen geringen Lernaufwand und unterstützt das Erlernen neuer Funktionen.

Regelsammlungen: mehr oder weniger umfangreiche, meist frei verfügbare Sammlungen von Regeln, um die Entwicklung von User Interfaces zu optimieren. Darunter fallen allgemeine **Usability-Prinzipien**,

z. B. Niensens *Usability Heuristics* [Nielsen 93] oder die *Google User Experience Principles* [Google 2009] sowie konkrete Regeln zu einem bestimmten Anwendungsgebiet. So gibt es „Do's and Don'ts“ für mobile Anwendungen oder „goldene Regeln“ für gutes Webdesign. Diverse Sammlungen sind als Bücher erhältlich, z. B. [Johnson 07].

User Interface Patterns: der Versuch, häufig auftretende oder ähnliche Designprobleme in Mustern zu beschreiben und bewährte Lösungsansätze zu bieten. Mittlerweile sind im GUI-Bereich einige hilfreiche Patterns-Sammlungen zu finden, z. B. [Tidwell 05]. Typischerweise sind die beschriebenen Patterns auf eine bestimmte GUI-Technologie bezogen. Die Abgrenzung zu Elementsammlungen, wie sie häufig in Styleguides verwendet werden, ist in der Praxis deshalb oft nicht eindeutig.

Hersteller- oder plattformabhängige Styleguides: beschreiben das vorgesehene Look&Feel der Applikationen eines bestimmten Betriebssystems mit dem Ziel einer konsistenten Anwendung aller GUI-Elemente wie Eingabefelder, Listboxen, Schaltflächen etc. Gute und nützliche Beispiele sind die Apple Human Interface Guidelines [Apple 00-09], die Windows User Experience Interaction Guidelines [Microsoft 05-09] und die Nokia Usability Documentation [Nokia 10].

Unternehmens-Styleguides: Vorgaben, welche die verschiedenen Applikationen eines Unternehmens bezüglich Look&Feel sowie Corporate Design erfüllen sollen. Dabei ist zu unterscheiden, ob es sich um Richtlinien für die firmeninterne Applikationslandschaft handelt oder um Anwendungen bzw. Produkte für externe Kunden. Auf den Einsatz von Unternehmens-Styleguides wird im Abschnitt „Usability-Standardisierung“ (S. 89) noch näher eingegangen.

Projekt-Styleguides: Richtlinien, welche die Konsistenz der Benutzerschnittstelle bei der Entwicklung einer Applikation (z. B. beim Einsatz verschiedener UI Designer) oder von Produkten für den Endkunden sicherstellen. In manchen Fällen müssen auch neue Bedienelemente definiert und beschrieben werden (z. B. für neuartige Consumer-Produkte).

Die folgende Abbildung zeigt ein Beispiel aus einem Styleguide. Zur Auswahl aus längeren Listen wird eine sortierbare Tabelle eingesetzt. Dieses GUI-Element sieht in jeder Anwendung gleich aus und verhält sich immer konsistent. Schrift und Kontrast wurden auf Lesbarkeit am Bildschirm optimiert, die Sortierfunktion ist intuitiv verständlich. Die Farbgebung entspricht dem Corporate Design des Unternehmens, die technische Umsetzbarkeit ist gewährleistet.

Zusammenstellung

	Column 1	Column 2 ▲	Column 3	Column 4
	45	Alfred ...	2.00	Abbcc ddd ...
	34	Beat ...	43.00	
+	156	Cora ...	22.00	
	4	Daniel ...	4'234.95	
	193	Emil ...	22.75	Kommentar ...
	13	Frank ...	123.00	Xxxx yyyy ...
	33	Gertrude ...	45.00	

- 1 Mittels Drag&Drop können die Zeilen umsortiert werden. Der Cursor verändert sich entsprechend.
- 2 Durch Klicken auf den Spaltenkopf wird die Tabelle auf-/ absteigend sortiert. Das Dreieck zeigt die aktuelle Sortierung.
- 3 „...“ wird dargestellt, wenn die Daten bearbeitet werden können. Bei Mouseover eine Schaltfläche darstellen.

Die Verwendung von Usability Guidelines

Das Wichtigste vorneweg: Selbst wenn Sie alle relevanten Regeln einhalten würden, könnten Sie noch immer eine für die Anwender unbrauchbare Lösung entwerfen. Mehr und mehr setzt sich die Erkenntnis durch, dass Zielgruppe und Nutzungskontext über die tatsächliche Qualität einer Benutzerschnittstelle entscheiden.

Usability Guidelines stellen in erster Linie ein Hilfsmittel für ein einheitliches und regelkonformes User Interface Design dar. Die Berücksichtigung von Richtlinien erleichtert dem Benutzer später die Anwendung, indem er auf bekannte Elemente trifft, die sich immer konsistent verhalten. Die Entscheidung, ob ein Regelwerk oder eine bestimmte Vorgabe für die vorliegende Anwendung und eingesetzte Technologie tatsächlich relevant ist, bedingt allerdings gute Fachkenntnisse. Die blinde Einhaltung von Vorgaben kann sonst schnell zu einem unnötigen „Klotz am Bein“ werden und gute Usability schlussendlich sogar verhindern.

Meist reicht es nicht aus, bestehende Regelsammlungen einzusetzen, weil diese nicht oder zu wenig auf die spezifische Situation zutreffen. Die Erstellung eines Styleguides, der die eingesetzten

GUI-Elemente definiert und beschreibt, ist in einem größeren Projekt ein wichtiges Instrument, um eine einheitliche Funktionsweise für die Anwender sicherzustellen. Zudem können zu einem frühen Zeitpunkt die Rahmenbedingungen für das User Interface Design gesetzt werden. Es ist zum Beispiel ein wesentlicher Unterschied, ob hoch effiziente Controls zur Unterstützung von Expertenbenutzern oder einfache, selbsterklärende Bedienelemente für Gelegenheitsbenutzer zum Einsatz kommen.

Gut erarbeitete, visuell und technisch abgestimmte Styleguides sind eine wertvolle Hilfe für die Entwickler. Statt das Rad jedes Mal neu zu erfinden, können sie ausgearbeitete Elemente verwenden, die den Ansprüchen bezüglich Ergonomie, Ästhetik, Corporate Design und technischer Umsetzbarkeit genügen.

Der Einsatz eines Styleguides in einem größeren Projekt oder Unternehmen hat auch eine organisatorische Komponente. Indem User-Interface-Elemente mit einem Namen vergeben, beschrieben und allen Beteiligten bekannt sind, können sowohl die Anforderungen an das User Interface als auch die technischen Restriktionen früher adressiert werden. Im Abschnitt „Usability-Standardisierung“ (S. 89) wird noch näher beleuchtet, wie der Einsatz von Styleguides den benutzerorientierten Prozess in einem Unternehmen unterstützen kann.

Die Problematik des Detaillierungsgrads

Je konkreter Usability Guidelines verfasst sind, umso enger wird deren Gültigkeitsbereich. Es kann zum Beispiel sinnvoll sein, detaillierte Styleguides für eine bestimmte Applikation zu erstellen, welche die Benutzeroberfläche genau beschreiben. Diese Regeln sind allerdings nicht einfach auf andere Anwendungen übertragbar. Umgekehrt kann es nützlich sein, generelle Usability-Richtlinien zu verfassen, die weitgehend technologieneutral und anwendungsübergreifend gelten. Wie diese Richtlinien im konkreten Fall dann allerdings eingehalten werden, bleibt dem jeweiligen User Interface Designer oder Entwickler überlassen.

Häufig entstehen Probleme, wenn zu detaillierte Styleguides mit einer konkreten Anwendung im Hinterkopf erstellt werden. Die Richtlinien werden in Unternehmen oft „offiziell abgesegnet“ und gelten zukünftig für weitere Anwendungen als strikte Vorgabe, zum Beispiel für externe Entwicklungs-Teams. Dies führt dazu, dass unter Umständen Regeln eingehalten werden müssen, die im vorliegenden Fall gar

keinen Sinn ergeben oder sogar kontraproduktiv sind. Es ist deshalb genau abzuwägen, welcher Detaillierungsgrad in einem spezifischen Fall anzustreben ist. Eine Abbildung eines beispielhaften Benutzerdialogs mit Erklärungen ist vielfach effektiver als die ausschließliche Formulierung von Regeln. Ein Beispiel kann auch leichter abstrahiert und auf andere Anwendungssituationen übertragen werden.

Was sollte ein guter Styleguide regeln?

Als Hilfestellung für die Erstellung eines eigenen Styleguides oder als Referenz, falls Sie einen Styleguide in Auftrag geben müssen, ist es nützlich zu wissen, welche Informationen im Werk enthalten sein sollten:

- *Technologische Rahmenbedingungen* und *Zielgruppe*: Auf welche Systeme bezieht sich der Styleguide? Wer ist der Empfänger des Regelwerks?
- *Software-Ergonomie*: Allgemeingültige Regeln, die im konkreten Fall zu berücksichtigen sind (z. B. Anzahl Menü-Einträge) sowie auch Regeln bezüglich der spezifischen Zielgruppe und Anwendung (z. B. vollständig über die Tastatur bedienbare Dialoge).
- *Anwendungsregeln*: Welche Elemente werden für welche Situation verwendet? Dies gilt sowohl für die Basis-Elemente des Betriebssystems (z. B. wann werden Radio Buttons, wann Listboxes eingesetzt) als auch für zusammengesetzte Elemente (z. B. sortierbare Tabellen, Wizards) und insbesondere für neu definierte Elemente (z. B. Kalender-Pop-up zur Datumsauswahl).
- *Verhalten der GUI-Elemente*: Beschreibung der Reaktion des Systems (z. B. Selektion eines Eintrags, Deaktivierung von Controls).
- *Navigation*: Beschreibung der Navigationselemente (z. B. Einsatz von Menüs, Links, Buttons).
- *Visuelles Design*: Farbschema, Kontraste, Schriften, Layout, Abstände, Icons etc. Hier werden auch Corporate-Design-Aspekte referenziert.
- *Technische Umsetzbarkeit*: Hinweise auf die technische Umsetzung, z. B. Referenz auf verfügbare GUI-Komponenten.
- *Terminologie*: Begriffe und Bezeichnungen der Benutzeroberfläche, wie wird der Benutzer angesprochen, welche Fachbegriffe werden verwendet, Formulierung von Fehlermeldungen.
- *Tastaturbedienung*: Shortcuts, Kommandos, Tabulator-Reihenfolge, Default Buttons in einem Dialog.

Checkliste für den Einsatz von Styleguides

- Müssen für Ihre Anwendung bestimmte Normen oder gesetzliche Vorgaben bezüglich der User-Interface-Gestaltung eingehalten werden?
- Gibt es bestehende Regelwerke, z. B. Regelsammlungen oder proprietäre Styleguides als Hilfestellung?
- Lohnt sich die Entwicklung eines eigenen Projekt-Styleguides?
- Sind die eingesetzten Usability Guidelines auf einer angemessenen Detaillierungsebene?
- Werden im Styleguide Aspekte beschrieben, die eher in die Spezifikation gehören?
- Könnte durch den Einsatz eines durchgängigen Styleguides die Kommunikation in Ihrem Projekt oder im Unternehmen erleichtert werden?

In Kürze

Methode	Usability Guidelines und Styleguides
Resultate	Regelkonformes User Interface Design Einheitliche Funktionsweise für die Anwender Anforderungen an die Benutzeroberfläche adressiert Hilfe für die User-Interface-Entwickler
Vorgehen	Verwendung vorhandener Regelsammlungen oder Erarbeitung eines Projekt-Styleguides
Aufwand	Abhängig von Umfang und Detaillierungsgrad Sammlung der relevanten Usability Guidelines: 1–2 PT Erarbeitung eines vollständigen Projekt-Styleguides: nicht unter 10 PT Unternehmens-Styleguides: nicht unter 100 PT
Beteiligte	Analyst, User Interface Designer
Planung	Nach Aufnahme der Benutzeranforderungen, vor Implementierung der Benutzerschnittstelle

Auf dem Prüfstand: Usability Testing

Vermutlich haben Sie schon davon gehört, dass Software und Produkte in sogenannten *Usability Labs* geprüft werden können. Vielleicht hatten Sie sogar schon selbst dazu Gelegenheit, einem *Usability-Test* als Zuschauer oder Testperson beizuwohnen. Obwohl man Usability-Labore nun seit gut 20 Jahren auch im deutschen Sprachraum findet, ist deren Bekanntheitsgrad noch immer relativ klein, und die Wenigsten wissen, was sich bei einem solchen Test genau abspielt. Wir wollen deshalb den typischen Ablauf eines formalen Usability-

Tests aufzeigen, um danach auch kurz auf andere, weniger formale Methoden einzugehen.

Der formale Usability-Test

Zunächst müssen Auftraggeber und Testleiter das Ziel des Usability-Tests klären. Fachleute unterscheiden zwischen *formativer* Evaluation, die eine Verbesserung des geprüften Systems zum Ziel hat, und *summativer* Evaluation, die ein Produkt im Sinne einer Qualitätskontrolle zusammenfassend prüft.

Als Vorbereitung für eine Usability-Testserie stellen Testleiter und Auftraggeber die Aufgaben zusammen, die von den Testpersonen mit der zu prüfenden Applikation bearbeitet werden sollen. Um ein gewisses Maß an Vergleichbarkeit zu erreichen, sind diese Aufgaben für jede Testperson dieselben. Man spricht deshalb auch von *Standardaufgaben*. Die Qualität der Ergebnisse eines Usability-Tests hängt wesentlich von der Ausarbeitung dieser Aufgaben ab. Im Kapitel „Einführung“ (S. 1) wurde aufgezeigt, dass die Usability einer neuen Lösung davon abhängt, wie gut die Abläufe für die Benutzer unterstützt werden. Die Erarbeitung relevanter und aus Benutzersicht realistischer Aufgaben sollte deshalb mit großer Sorgfalt durchgeführt werden. Wurden bereits Anwendungsszenarien erstellt, dienen diese als Grundlage (siehe Abschnitt „Personas und Szenarien“, S. 28). Ein guter Testleiter wird auf die Einhaltung folgender Kriterien für die Standardaufgaben achten, die Sie als Auftraggeber kontrollieren sollten:

- Die Aufgabenstellung ist ein aus Benutzersicht realistisches Szenario und könnte sich tatsächlich so abspielen.
- Es wird ein Ziel aus Sicht des Anwendungsgebietes formuliert, keine technische Anleitung zur Erfüllung dieses Zieles. Zum Beispiel: „Sie suchen nach einer passenden Farbe für ...“ ist besser als „setzen Sie die Filterkriterien auf gelb und rosa“.
- Die Aufgaben stellen für die Testpersonen einen mittleren Schwierigkeitsgrad dar. Sie sollten lösbar, jedoch nicht zu trivial sein.
- Begriffe und Bezeichnungen, welche in der Applikation vorkommen, sind zu vermeiden. Zum Beispiel: „Sie überweisen den Betrag von ...“ ist neutraler als „gehen Sie im Menü auf Einzahlungen“.

Neben den Aufgaben für den Test muss auch das zu prüfende System selbst bzw. der Prototyp entlang der beabsichtigten Anwendungsszenarien vorbereitet werden. Dies kann zum Beispiel bedeuten, dass bestimmte Systemzustände oder Ausgaben abgebildet werden müssen,

um der Testperson den Eindruck eines möglichst realistischen, lauffähigen Systems zu vermitteln.

Für eine Usability-Testserie sollten Testpersonen eingeladen werden, die möglichst aus der Benutzergruppe der zu prüfenden Applikation stammen, d. h. tatsächlich zu den späteren Benutzern gehören oder gehören könnten. Während dies bei Consumer-Produkten oder Internet-Anwendungen vielleicht auch die Sekretärin oder die Mitarbeiter aus dem Nachbarbüro sein können, ist es für spezialisierte Anwendungen unerlässlich, die entsprechenden Fachleute zu rekrutieren.

Die notwendige Anzahl Testpersonen hängt im Wesentlichen von den Zielen des Tests ab. In der Regel genügen fünf bis sieben Testpersonen, um die wichtigsten Anwendungsszenarien einer Applikation anhand eines Prototyps zu prüfen und gezielte Verbesserungsmaßnahmen einzuleiten. Sollen dagegen mit großer Sicherheit sämtliche Stolpersteine vor der Einführung einer Anwendung ausgeräumt werden, sind weitere Testserien notwendig (mehr dazu im Abschnitt „Wie viele Testpersonen sind notwendig?“, S. 66).

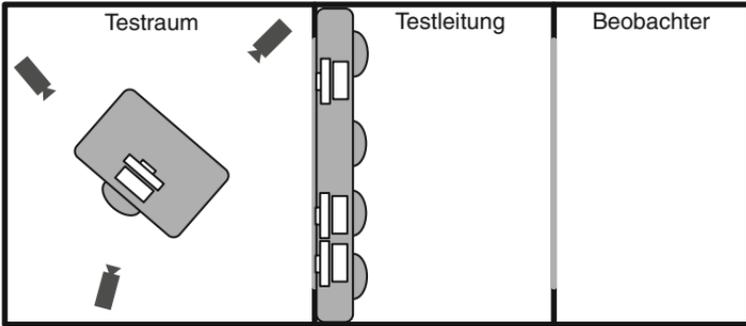
In manchen Fällen kann es sinnvoll sein, dass die Testpersonen eine kurze Schulung oder Einführung in die Applikation erhalten, zum Beispiel wenn es sich um Expertenbenutzer handelt, die nach einer Einarbeitsphase eine hocheffiziente Applikation bedienen sollen. Handelt es sich dagegen um Gelegenheitsbenutzer, etwa für eine Consumer-Anwendung, sollte die Applikation selbstverständlich nicht zuerst erklärt werden.

Der Testleiter führt jede Testperson in die Ziele und den Ablauf des Usability-Tests ein, und es werden Spielregeln vereinbart:

- Die Testperson darf den Test jederzeit unterbrechen bzw. abbrechen.
- Die Testperson wird meist gebeten, laut zu denken, d. h. ihre Arbeit für die Beobachter zu kommentieren.
- Sollte die Testperson mit einer Aufgabe nicht mehr weiterkommen, kann sie selbstständig zur nächsten Aufgabe weitergehen.
- Die Beobachter melden sich nur, wenn wirklich notwendig. Sie sollten den Testverlauf möglichst nicht beeinflussen. Eine Konversation erfolgt kontrolliert, zum Beispiel über eine Gegensprechanlage.

Während der eigentlichen Testphase arbeitet die Testperson in einem speziell eingerichteten Testraum mit der zu prüfenden Applikation

gemäß den Aufgaben. Die folgende Abbildung zeigt den Aufbau eines typischen Usability Labs:



Als Beobachter sollten Vertreter des Auftraggebers, Entwickler und Usability-Experten teilnehmen. Testleiter und Beobachter verfolgen die Arbeit aus einem separaten Raum, der häufig durch eine Glasscheibe abgetrennt ist. Der Bildschirm bzw. das zu prüfende Produkt sowie häufig auch das Gesicht der Testperson und die Situation als Ganzes werden auf Video aufgezeichnet. Die folgende Abbildung zeigt eine solche Sicht:

Die Beobachter protokollieren unklare und problematische Situationen oder Fehler, die sich in der Anwendung mit der Applikation ergeben. Die Testphase sollte nicht länger als etwa eine Stunde dauern.

In einer Nachbesprechung analysieren die Beobachter die entsprechenden Stellen im Video nochmals mit der Testperson. Dabei werden bereits erste Möglichkeiten für Verbesserungsmaßnahmen besprochen. In jedem Fall sollte die Testperson dazu Gelegenheit haben, das Erlebte in einem kurzen Interview oder frei zu kommentieren. Oft ergeben sich gerade aus eher informellen Situationen wertvolle Hinweise.

Die Feststellungen und Verbesserungsvorschläge werden in einem Testbericht festgehalten. Die Resultate hängen einerseits von der



Die 7±2 wichtigsten Usability-Methoden

Erfahrung von Testleiter und Beobachter ab und andererseits von der Zeit, die für die Ausarbeitung von Verbesserungsvorschlägen zur Verfügung steht. Es ist deshalb lohnenswert, das Ergebnis vorher zu vereinbaren. Wiederum gibt es einige formale Kriterien, die ein guter Testbericht erfüllen sollte:

- Gute wie schlechte Ergebnisse werden aufgeführt.
- Für jede Schwachstelle bzw. Verbesserungsmaßnahme wird ein Schweregrad angegeben.
- Die Schwachstellen sind mit Screenshots illustriert.
- Die Standardaufgaben sowie eine Beschreibung der Testpersonen werden angefügt.
- Es wird klar zwischen beobachteten Problemen und persönlichen Meinungen und Vorschlägen unterschieden.

Stärken und Schwächen des Usability-Tests

Lange Zeit galt der formale Usability-Test als die Königsmethode im Usability Engineering, und tatsächlich hat die Methode auch einige Stärken:

- Unter Laborbedingungen können Schwachstellen der Benutzeroberfläche eindeutig nachgewiesen werden.
- Methodische Gütekriterien wie Objektivität, Reliabilität und Validität können weitgehend eingehalten werden (siehe auch Abschnitt „Methodische Gütekriterien“, S. 65).
- Die Beobachtungssituation in einem Usability Lab ist optimal, die Beobachter können sich ein gutes Bild über die Stärken und Schwächen einer neuen Lösung machen.
- Die Beteiligten werden dazu gezwungen, einem Benutzer zuzusehen, ohne in das Geschehen einzugreifen. Jeder Usability-Testleiter wird Ihnen versichern, dass die Scheibe zwischen Test- und Beobachtungsraum vor allem dazu da ist, die Beobachter davon abzuhalten, dem Benutzer zu helfen.
- Schwierigkeiten in der Anwendung werden schnell deutlich.
- Die Methode ist für alle Beteiligten gut sichtbar. Die Bedeutung von Usability Engineering wird unmittelbar ersichtlich.

Die Schwächen der Methode sind die folgenden:

- Eine formale Usability-Testserie ist aufwändig. Es muss genügend Zeit für die Ausarbeitung der Aufgaben, die Vorbereitung des zu prüfenden Systems, die Rekrutierung der Testpersonen, die Durchführung der Testreihe in einem Usability Lab und nicht zuletzt

die Erarbeitung von Verbesserungsvorschlägen und das Verfassen eines Testberichts eingesetzt werden.

- Die Methode kommt relativ spät zum Einsatz. Prototypen müssen so weit vorbereitet sein, dass die Testpersonen selbstständig damit arbeiten können.
- Es besteht die Gefahr, dass der Testbericht mit wertvollen Resultaten nach dem Test in der Schublade des Auftraggebers verschwindet, weil Kenntnisse oder Mittel für Verbesserungsmaßnahmen fehlen. Es ist deshalb notwendig, dass Testleiter, Auftraggeber und Entwickler die Schwachstellen zusammen besprechen und priorisieren. Es kann unter Umständen sogar sinnvoll sein, auf einen ausführlichen Bericht zu verzichten und die Zeit direkt in Lösungsvorschläge zu investieren.

Hintergrund: Methodische Gütekriterien

In der Testtheorie verwendet man eine Reihe von Gütekriterien, anhand derer Aussagen über die Qualität eines Messinstruments in einer bestimmten Untersuchung gemacht werden. Die Kenntnis dieser Gütekriterien ist auch für die Anwendung von Usability-Methoden von Relevanz, besonders wenn es um die Evaluation von Systemen geht. Als Hauptgütekriterien werden unterschieden:

- *Objektivität*: Die Resultate der Untersuchung sollen unabhängig von den Rahmenbedingungen sein. Ein wichtiger Aspekt ist dabei die Unabhängigkeit von der durchführenden Person, die als *Testleiterunabhängigkeit* bezeichnet wird.
- *Reliabilität*: Bei wiederholter Anwendung unter gleichen Bedingungen soll das Instrument zu gleichen Resultaten kommen.
- *Validität*: Das Instrument soll das messen, was es vorgibt, in unserem Fall also die Usability eines Systems und nichts anderes. Dies kann zum Teil durch die Ausschaltung von ungewollten Einflüssen erreicht werden (*interne Validität*). Von Bedeutung ist indessen auch die Generalisierbarkeit der Ergebnisse (*externe Validität*). Sie kann durch eine Erhöhung der Anzahl befragter Personen und eine Überprüfung mit anderen Methoden gesteigert werden.

Alternativen zum formalen Usability-Test

Der **Usability Walkthrough** ist eine Alternative zum formalen Usability-Test. Statt eine Testperson unter kontrollierten Bedingungen in einem separaten Testraum allein arbeiten zu lassen, begleitet der Testleiter den Benutzer und moderiert den Testablauf. Wie beim formalen Test bearbeitet der Benutzer realistische Aufgaben mit dem zu

prüfenden System, aber der Moderator hat die Möglichkeit, direkt einzugreifen, Fragen zu stellen und bestimmte Abläufe mit dem Benutzer durchzugehen. Dies kann, muss aber nicht in einem Usability Lab durchgeführt werden. Diese Methode eignet sich besonders gut, um früh im Prozess noch unfertige Prototypen zu evaluieren, ohne dass schon ein lauffähiges System vorhanden wäre. Es ist selbstredend, dass die Testleiterunabhängigkeit bei dieser Spielart nicht mehr gegeben ist und der Moderator sehr genau wissen muss, wie er den Benutzer anleiten kann, ohne ihn zu stark zu beeinflussen.

Als **mobiles Usability Lab** wird eine mobile Einrichtung zur Durchführung von Usability-Tests vor Ort anstatt in einem Labor bezeichnet. Statt das User Interface auf Video aufzuzeichnen, verwenden heute viele Testleiter spezielle Programme, die eine Aufzeichnung der Benutzerinteraktion auf dem Testcomputer selbst auf Harddisk erlauben. Ausgestattet mit einem Laptop und einer Webcam lassen sich auf diese Weise Usability-Testserien ohne die Kosten eines Usability Labs durchführen. Nutzen und Resultate hängen aber stark von der Erfahrung des Testleiters ab.

Mobile Usability-Tests sind vor allem dann angebracht, wenn bei den Benutzern vor Ort getestet werden soll, zum Beispiel an speziellen Arbeitsplätzen oder wenn die Umgebung einen maßgeblichen Einfluss hat, beispielsweise bei öffentlichen Terminals wie Ticketautomaten in Bahnhöfen oder Bankomaten im Freien. Fachleute sprechen dann auch von *Usability-Feldtests*.

Wie viele Testpersonen sind notwendig?

Stellen Sie sich vor, Sie wären ein Ladenbesitzer. Gerade haben Sie beobachtet, wie eine ältere Dame im Eingangsbereich über eine hervorstehende Türschwelle stolpert. Kurz darauf geschieht einem jungen Mann dasselbe. Würden Sie die Schwelle ersetzen oder warten, bis weitere Personen straucheln?

Kritische Usability-Probleme können ebenso deutlich sein, wenn sie im Rahmen eines Usability-Tests auftreten. Wenn zwei oder drei Testpersonen an derselben Stelle der Anwendung Mühe bekunden, ist es offensichtlich, dass ein Stolperstein vorliegt, den es auszuräumen gilt. Es ist schlichtweg nicht notwendig, eine vierte oder fünfte Testperson dabei zu beobachten. Mit anderen Worten: Um mittels Usability-Tests Verbesserungen der Benutzerschnittstelle zu erreichen, genügen qualitative Aussagen. Es sind keine quantitativen Untersuchungen mit vielen Benutzern erforderlich (siehe auch Abschnitt „Hintergrund“:

quantitative und qualitative Methoden“, S. 68). Trotzdem stellt sich die Frage, wie viele Testpersonen sinnvoll und notwendig sind, um genügend Sicherheit zu erhalten, dass kritische Stellen auch aufgedeckt werden. Im Folgenden finden Sie einige Anhaltspunkte:

- Iteratives Prototyping, Verbesserung und Anpassung auf die Benutzerbedürfnisse, qualitative Aussagen: 4–6 Benutzer pro Iteration. Systeme mit hohen Anforderungen an die Benutzbarkeit: 7–15 Benutzer. Risikominimierung bei kritischen Systemen: nicht unter 15 Testpersonen.
- Qualitätskontrolle vor Einführung eines Systems, quantitative Aussagen: abhängig von Systemumfang und Anforderungen an die Benutzbarkeit: nicht unter 10 Testpersonen.

In Kürze

Methode	Usability Testing
Resultate	User Interface auf Usability geprüft Schwachstellen identifiziert, dokumentiert und priorisiert Verbesserungsmaßnahmen erarbeitet
Vorgehen	Testen mit Benutzern anhand von Standardaufgaben
Aufwand	Formaler Usability-Test im Labor mit 5–7 Benutzern (Vorbereitung, Durchführung und Auswertung): Testleiter 10 PT, Auftraggeber 5–10 PT, Testpersonen total 3–5 PT Usability Walkthrough mit 5–7 Benutzern: Testleiter 5 PT, Auftraggeber 3–5 PT, Testpersonen total 2–3 PT
Beteiligte	Auftraggeber, Entwickler, Benutzer, Testleiter, Beobachter
Planung	Nach der Erstellung eines ersten Prototyps Am Ende jeder Iteration Vor der Einführung

Zahlenmaterial: Fragebögen

Beziehen Sie regelmäßig die neusten Studien bekannter Analysten und Marktforschungsinstitute? Für die Erhebung des zugrunde liegenden Zahlenmaterials werden häufig Fragebögen benutzt. Die Methodik für die Erstellung und den Einsatz von Fragebögen hat ihre Wurzeln in den Sozialwissenschaften, wo statistisch auswertbare Daten von Personen gefragt sind, wie etwa bei der Erhebung von Einstellungen, Meinungen und Erfahrungen, oder wenn die Ergebnisse mit einer größeren Gesamtheit verglichen werden sollen, etwa bei psychologischen Studien.

Im Usability Engineering sind Benutzerbefragungen mittels Fragebögen eine wichtige Methode, um Antworten von einer größeren

Anzahl Personen zu erhalten. Wenn Sie schon selber einen Fragebogen entworfen haben, zum Beispiel um Meinungen von Mitarbeitern oder Kunden abzufragen, dann wissen Sie, dass dies gar nicht so einfach ist. Die Zuverlässigkeit der Aussagen hängt von der Qualität des Fragebogens, der Auswahl der befragten Personen und der richtigen Durchführung der Befragung ab.

Einsatz von Benutzerbefragungen

Erinnern Sie sich an die im Abschnitt „Usability-Methoden im Zusammenhang“ (S. 14) vorgestellten Aufgabenbereiche im Usability Engineering? Benutzerbefragungen dienen zwei Aufgaben: Sie können sowohl zur Analyse von Benutzern und Kontext als auch zur Beurteilung der Usability eines Systems (Evaluation) eingesetzt werden. Benutzerbefragungen stellen eine Ergänzung zu den bereits vorgestellten Usability-Methoden dar. Der Analyst erreicht damit im Vergleich zu Contextual Inquiry oder Usability-Tests eine größere Anzahl Personen. Fragebögen dienen ihm zur standardisierten Erfassung und Auszählung der Antworten.

Abhängig vom Ziel und der Fragestellung können verschiedene Instrumente eingesetzt werden. Die Palette reicht von einfachen, selbst erstellten Fragebögen bis zum Einsatz von methodisch geprüften Usability-Standardfragebögen.

Für die Durchführung einer Benutzerbefragung kommen verschiedene Befragungsformen in Betracht: Ein Fragebogen kann beispielsweise schriftlich verschickt, online angeboten oder per Telefonbefragung ausgefüllt werden.

Der Einsatz von Fragebögen erfolgt oft mit dem Ziel, für die gesamte Benutzergruppe repräsentative Aussagen zu erhalten. Da solche Untersuchungen auf zählbare Werte abzielen, sprechen Fachleute auch von *quantitativen*, im Gegensatz zu *qualitativen* Studien. Für sie gilt eine Reihe von methodischen Besonderheiten, die beachtet werden müssen. Da dies mit entsprechendem Aufwand verbunden ist, sollte genau geprüft werden, ob die vorliegende Fragestellung überhaupt mit einer quantitativen Befragung beantwortet werden kann.

Hintergrund: quantitative und qualitative Methoden

Die empirische Forschung unterscheidet zwei grundsätzliche Ansätze zur Erkenntnisgewinnung:

- *Quantitative* Studien haben zum Ziel, zahlenmäßige Ausprägungen möglichst genau zu beschreiben. Dabei wird üblicherweise eine repräsentative

Stichprobe befragt und die erhobenen Daten auf die Grundgesamtheit verallgemeinert. In der Regel wird vorher eine Hypothese festgelegt, die anhand der Ergebnisse überprüft werden soll. Quantitative Methoden laufen typischerweise stark standardisiert ab, um möglichst gleiche Voraussetzungen für die Aussagen der Befragten zu schaffen.

- **Qualitative** Erhebungen zielen darauf ab, Hintergründe, Zusammenhänge und Ursachen festzustellen. Dabei wird auf die subjektiven Aussagen der Befragten Wert gelegt. Der Ablauf dieser Methoden ist im Vergleich zu quantitativen Verfahren flexibel, offen und explorativ. Oft werden während der Durchführung neue Hypothesen generiert, um diese in einer nächsten Iteration weiterzuverfolgen. Aus qualitativen Daten kann man keine Mengenangaben ableiten.

Die folgende Tabelle stellt die beiden Ansätze gegenüber:

Quantitative Forschung

Viele Teilnehmer
Repräsentative Stichprobe
Hypothesen prüfen
Standardisiert
Zahlenmäßige Ausprägungen
Geschlossene Fragen
Statistische Analyse
Einfache Auswertung
Beispiele: schriftliche Befragung,
standardisiertes Telefon-Interview

Qualitative Forschung

Wenige Teilnehmer
Typische Vertreter
Hypothesen bilden
Flexibel, explorativ
Hintergründe, Zusammenhänge
Offene Fragen
Inhaltsanalyse
Aufwändige Auswertung
Beispiele: Contextual Inquiry, Stakeholder
Interview, Focus Group

In keinem Fall darf ein Fragebogen als eine einfache Sammlung von Fragen verstanden werden, die man mal eben so verschickt. Die Durchführung einer Befragung setzt voraus, dass sich der Ersteller genau überlegt, welche Fragestellung beantwortet werden soll, wie die Untersuchung durchgeführt wird und wie der Fragebogen dafür aufgebaut sein muss. Methodiker sprechen auch vom *Untersuchungsdesign* und der *Fragebogenkonstruktion*.

Planung einer Benutzerbefragung

Zur Festlegung eines geeigneten **Untersuchungsdesigns** gehören einige zentrale Überlegungen:

- Welche Fragen oder Hypothesen sollen beantwortet werden? Werden reine Fakten erhoben, soll ein System beurteilt oder ein Vergleich durchgeführt werden?

Die 7±2 wichtigsten Usability-Methoden

- Wie ist der zeitliche Ablauf der Untersuchung? Wird nur einmal erhoben, werden die Aussagen verschiedener Gruppen verglichen oder werden die gleichen Benutzer in Abständen mehrmals befragt?
- Wie erfolgt die Auswahl der zu befragenden Benutzer: als zufällige Stichprobe oder nach bestimmten Kriterien?
- Wie viele Personen müssen befragt werden, um statistisch genügend gesicherte Aussagen machen zu können?
- Mit welchen Instrumenten wird die Untersuchung durchgeführt? Wird ein bestehender Fragebogen eingesetzt oder wird dieser erstellt?

Fragebogenkonstruktion

Konstruktion und Einsatz eines Fragebogens müssen methodisch korrekt erfolgen, um aussagekräftige Ergebnisse zu erhalten. Dies gilt auch für einfache Umfragen mit wenigen Fragen. Bevor die Erstellung eines eigenen Fragebogens ins Auge gefasst wird, sollte deshalb abgeklärt werden, ob nicht Standardfragebögen für die Fragestellung verfügbar sind. Für die Konstruktion von eigenen Fragebögen zu statistisch zuverlässigen Aussagen lohnt sich in jedem Fall das Hinziehen eines Fragebogen-Experten.

Die folgenden methodischen Aspekte sind sowohl für die Erstellung eines eigenen Fragebogens als auch bei der Auswahl eines Standardfragebogens relevant:

- Soll mit offenen oder geschlossenen Fragen gearbeitet werden? Bei offenen Fragen können die Benutzer freie Antworten formulieren, bei geschlossenen wird aus vorgegeben Antworten ausgewählt. Offene Fragen erlauben es, eine Frage in der Breite auszuleuchten; sie sind allerdings in der Auswertung aufwändiger als geschlossene Fragen. Offene Fragen sind deshalb für qualitative Untersuchungen besser geeignet, während geschlossene Fragen eher für quantitative Studien verwendet werden.
- Kommen Skalen (z. B. Werte von 1–7) zum Einsatz? Was bedeuten die Skalen (z. B. einverstanden – nicht einverstanden, Note 1–6)? Dies ist von Bedeutung im Hinblick auf die Auswertung und Interpretation der erhobenen Werte.
- Wie erfolgt die Instruktion zum Ausfüllen des Fragebogens?
- Sind alle Fragen für die Zielgruppe verständlich? Ein Fragebogen sollte immer an einer Versuchsstichprobe getestet werden.

- Wie lange dauert das Ausfüllen des Fragebogens? Die Abbruchquote steigt, und die Antwortqualität sinkt mit der Länge eines Fragebogens. Diese Effekte sind bei Online-Befragungen noch ausgeprägter als bei schriftlichen Fragebögen.

Für die methodischen Details zur Durchführung von Fragebogen-Untersuchungen sei an dieser Stelle auf die einschlägige Fachliteratur verwiesen, z. B. [Bortz 06].

Analyse von Benutzern und Kontext

Fragebögen können zur Analyse der Benutzer und des Kontexts der Anwendung eingesetzt werden und damit zur Klärung der Anforderungen an eine neue Lösung beitragen. Da es in der **Anforderungsanalyse** meist darum geht, ein neues Gebiet in seiner Breite auszuloten, Details der täglichen Arbeit zu erfassen oder Ursachen und Zusammenhänge zu erkunden, ist dies zunächst einmal die Domäne von qualitativen Methoden. Ergänzend zur Durchführung von Contextual Inquiry, Interviews oder Beobachtungen kann es indessen erwünscht sein, bestimmte Aspekte mit einem größeren Benutzerkreis abzuklären oder Aussagen gezielt zu erhärten. Eine sinnvolle Anwendung wäre beispielsweise, die Benutzer zunächst mit offenen Fragen nach Vorlieben und Problemen mit bestehenden Systemen zu fragen und damit erste Hinweise für Verbesserungsmaßnahmen zu sammeln und diese in einem zweiten Schritt mittels einer Beurteilungsskala bewerten zu lassen.

Fragebögen können auch eingesetzt werden, um Fakten über Benutzer und Anwendung zu erheben, die im Hinblick auf die Anforderungen an die neue Lösung relevant sind:

- Alter, Geschlecht, Ausbildung und Erfahrung der Benutzer
- Rollen, Aufgaben, Tätigkeiten
- Häufigkeit, zeitliche und örtliche Verteilung der Anwendung
- Vorhandene technische Ausstattung, z. B. Betriebssystem, Bildschirmgröße und -auflösung, Browserversion, vorhandene Applikationen usw.

In Marktbefragungen für neue Produkte wird manchmal auf diese Weise versucht, Auskunft von den zukünftigen Benutzern über die erwünschte Funktionalität oder die Notwendigkeit bestimmter Features zu erhalten. Die Möglichkeiten solcher Befragungen sind allerdings begrenzt. Können sich die befragten Personen beispielsweise nicht ganz genau vorstellen, worum es bei der neuen Lösung geht, sind die

Die 7±2 wichtigsten Usability-Methoden

Antworten irreführend oder wertlos. Auch die Schwierigkeiten, die sich aus den gewünschten Funktionen ergeben, können die Benutzer kaum abschätzen. Sie entstehen erst in der Anwendungssituation.

Beurteilung der Usability: Standardfragebögen

Usability-Standardfragebögen sind Instrumente zur Beurteilung der Usability von Systemen, die gekauft werden können oder sogar frei verfügbar sind. Sie können mit relativ wenig Aufwand zur Evaluation von lauffähigen Prototypen, zur Identifikation von Schwachstellen oder als Qualitätskontrolle bei produktiven Systemen eingesetzt werden.

Die Beurteilung durch die Benutzer erfolgt dabei nach bestimmten Kriterien. Verbreitete Standardfragebögen zur Beurteilung der Usability von Software sind der IsoMetrics [Willumeit, Gediga & Hamborg 96], der in der aktuellen Version 75 Fragen (Items) umfasst, und der ISONORM 9241/10 [Prümper & Anft 97] mit insgesamt 35 Fragen. Beide Fragebögen verwenden die Dialogkriterien der ISO-Norm 9241-10 (siehe auch Abschnitt „Usability Guidelines und Styleguides“, S. 54).

Der ISONORM 9241/10 wurde auch in eine Online-Befragung umgesetzt [Richter 99] und stellt damit ein einfaches Mittel zur Beurteilung von Internet-Anwendungen dar. Die folgende Abbildung zeigt einen Ausschnitt dieser Online-Version:

Unterstützt die Software die Erledigung Ihrer Arbeitsaufgaben, ohne Sie als Benutzer unnötig zu belasten?

Die Software	---	--	-	+/-	+	++	+++	
ist kompliziert zu bedienen	●	●	●	●	●	●	●	ist unkompliziert zu bedienen
bietet nicht alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen	●	●	●	●	●	●	●	bietet alle Funktionen, um die anfallenden Aufgaben effizient zu bewältigen

Die Bearbeitung des Fragebogens dauert etwa 20 Minuten. Die Auswertung erfolgt im Sinne einer Gesamtbeurteilung der Usability über die Durchschnittswerte aller Fragen oder auf Ebene der einzelnen ISO-Kriterien. So lassen sich beispielsweise erste Hinweise auf Schwachpunkte finden. Die folgende Tabelle zeigt ein Beispiel

einer Auswertung mit Mittelwert x und Standardabweichung s (Werte 1–7):

ISO-Kriterium	x	s
Aufgabenangemessenheit	4,76	0,88
Selbstbeschreibungsfähigkeit	5,20	0,87
Steuerbarkeit	4,64	1,05
Erwartungskonformität	4,95	0,96
Fehlertoleranz	4,76	0,97
Individualisierbarkeit	3,76	1,22
Lernförderlichkeit	5,48	1,06
ISO-Gesamtbeurteilung	4,80	0,75

Ein wichtiger Aspekt bei Standardfragebögen ist die Vergleichbarkeit der Daten. Dadurch, dass jede Person dieselben Urteilsskalen ausfüllt, können die Ergebnisse verdichtet, statistisch ausgewertet und die Aussagen miteinander verglichen werden. Standardfragebögen sind deshalb ein geeignetes Mittel, um Vergleiche anzustellen, zum Beispiel wenn die Benutzbarkeit zweier verschiedener Prototypen, die Aussagen von unterschiedlichen Benutzergruppen oder die Beurteilung eines Systems zu verschiedenen Zeitpunkten untersucht werden soll.

Für die Interpretation der Resultate ist es wichtig zu wissen, dass bei der Beurteilung mittels Fragebögen folgende generelle *Urteilsfehler* auftreten können:

- Bei der Beurteilung der Software besteht die Gefahr, dass die Beurteiler nicht zwischen den einzelnen Kriterien differenzieren, sondern sich von ihrem Gesamteindruck der Software beeinflussen lassen (*Halo-Effekt*).
- Das zu beurteilende Objekt kann systematisch zu niedrig oder zu hoch eingestuft werden (*Milde-Härtefehler*). Die Gründe dafür können vielfältig sein und beispielsweise mit Ablehnungen oder Vorlieben zu tun haben.
- Es besteht eine gewisse Tendenz, alle Kriterien im mittleren Bereich einer Urteilsskala einzustufen (*zentrale Tendenz*). Ist dieser Effekt ausgeprägt, kann dies ein Hinweis auf eine mangelnde Kenntnis des zu beurteilenden Objekts sein.

Kann man Usability messen?

In der Praxis taucht hin und wieder die Frage auf, ob und wie man die Usability von Systemen oder Produkten messen kann. Damit ließen sich unterschiedliche Produkte miteinander vergleichen oder Qualitätsprüfungen durchführen. Nicht zuletzt wäre eine hohe, mit anerkannten Instrumenten gemessene Usability auch ein gutes Verkaufsargument. Nicht selten werden Vergleiche publiziert, die auf Ergebnissen aus Usability-Tests oder Expertenmeinungen basieren. Genügt das? Wie müsste ein solches Instrument zur Messung der Usability aussehen?

Halten wir uns nochmals die Definition aus dem Kapitel „Einführung“ (S. 1) vor Augen: Ein Mensch-Computer-System setzt sich zusammen aus dem Benutzer, dem Werkzeug (System), der Aufgabe und dem Umfeld. Eine ganzheitliche Bewertung der Usability ist nur unter Berücksichtigung aller vier Komponenten möglich, sollte also mit tatsächlichen Benutzern in deren Umfeld stattfinden. Weiter muss eine genügend hohe Anzahl Fälle vorhanden sein, um statistisch signifikante Aussagen machen zu können. Ein gutes Messinstrument muss außerdem den testtheoretischen Gütekriterien genügen (siehe auch Abschnitt „Methodische Gütekriterien“, S. 65). Und schließlich muss für die Interpretation genügend Vergleichsmaterial verfügbar sein, um überhaupt eine Aussage über die Höhe der gemessenen Werte machen zu können („das Produkt erreicht den Wert 6,5“).

Qualitative Evaluationsmethoden wie Usability-Tests oder Experten-Reviews liefern wichtige Ergebnisse für die Entwicklung und interessante Vergleiche, was in den meisten Fällen auch genügt. Für eine *Messung* von Usability sind sie hingegen nicht geeignet. Entsprechende quantitative Untersuchungen mit Benutzern in ihrem realen Umfeld wären vorstellbar, sind aber zum einen sehr aufwändig und zum anderen nicht über verschiedene Systeme vergleichbar.

Ein vielversprechender Ansatz ist der Einsatz von Usability-Standardfragebögen, die als Evaluationsinstrument die oben genannten Anforderungen weitgehend erfüllen. Die Beurteilung eines Systems findet aus Sicht der Benutzer im Anwendungskontext statt und erfolgt einheitlich über dieselben Kriterien. Die Fragebögen wurden von ihren Autoren bezüglich Gütekriterien überprüft und optimiert. Mit einer Normierung streben die Autoren zudem an, eine Vergleichsbasis für verschiedene Anwendungen zu schaffen. So wäre es zumindest theoretisch möglich, auch unterschiedliche Software-Produkte über

ihre Benutzer beurteilen zu lassen und die so erhaltenen Werte zu vergleichen.

Über den Nutzen eines Vergleichs unterschiedlicher Systeme oder Produkte lässt sich streiten. Eine interessante Anwendung für die Praxis ist indes sicher folgende: Mittels regelmäßiger Benutzerbefragungen lässt sich im Sinne einer Qualitätskontrolle die Benutzbarkeit eines Systems über die Zeit überprüfen. Dies ist beispielsweise bei Internet-Anwendungen mithilfe einer regelmäßig geschalteten Online-Befragung ohne großen Aufwand möglich. Sollten sich die Werte (z. B. nach einem Redesign) negativ verändern, ist Handlungsbedarf angesagt. Einige Anbieter nutzen diese Möglichkeit und überprüfen die Benutzbarkeit ihrer Produkte regelmäßig mittels Online-Fragebögen.

Checkliste zum Einsatz von Fragebögen

- Ist für die vorhandene Fragestellung eine quantitative Untersuchung notwendig und geeignet, oder sind vielmehr qualitative Methoden besser (z. B. Contextual Inquiry, Interviews, Usability-Tests mit Prototypen)?
- Dienen Zahlenwerte und statistische Angaben zur Beantwortung der Fragestellung?
- Können die Benutzer die Fragen aus ihrem Wissen oder ihrer Erfahrung beantworten?
- Lässt sich die Fragestellung so in einen Fragebogen verpacken, dass sie verständlich bleibt und in kurzer Zeit beantwortet werden kann? Wie groß ist die Gefahr von Falschantworten?
- Ist eine statistische Auswertung der Antworten möglich, sinnvoll und bezüglich Aufwand vertretbar?
- Muss ein Fragebogen selbst entwickelt werden oder ist ein geeigneter Standardfragebogen verfügbar?

In Kürze

Methode	Benutzerbefragung mittels Fragebögen
Resultate	Quantitative Daten, statistische Aussagen zur Usability eines Systems, als Ergänzung in der Anforderungsanalyse oder zur Qualitätssicherung
Vorgehen	Einsatz eines Standardfragebogens oder Erstellung eines zugeschnittenen Fragebogens, Durchführung der Befragung und Auswertung der Antworten

Die 7±2 wichtigsten Usability-Methoden

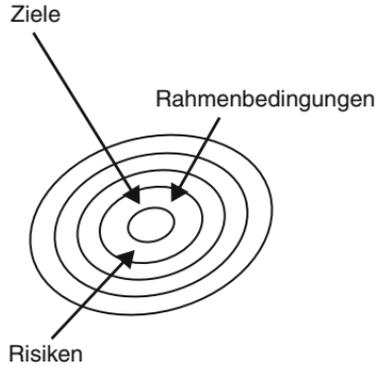
Aufwand	Erstellen und Auswerten eines kleineren qualitativen Fragebogens: 5–10 PT Durchführen und Auswerten einer Untersuchung mit Standardfragebogen: 10–20 PT Erstellen und Auswerten einer quantitativen Fragebogen-Studie: nicht unter 30 PT
Beteiligte	Auftraggeber, Fragebogen-Experte
Planung	Analyse: Im Anschluss an Contextual Inquiry Evaluationsinstrument: Im Rahmen eines Pilottests oder zur ständigen Qualitätssicherung

Usability im Griff: Planung

„Wir entwickeln ein neues Produkt für jedermann. Wie viel Usability brauchen wir denn?“

Wir denken, etwa 5 Kilo sollten reichen. Am besten scheinchenweise. Aber was genau haben Sie denn vor?

In den letzten Kapiteln haben wir die wichtigsten Usability-Methoden und ihren Zusammenhang in einem benutzerorientierten Prozess im Allgemeinen vorgestellt. Damit die Techniken richtig eingesetzt werden können, müssen die projektspezifischen Ziele, Risiken und Rahmenbedingungen berücksichtigt werden. Man benötigt einen Projektleiter, der die notwendigen Methoden einplant, gut ausgebildete Fachkräfte einsetzt und den Fortschritt überwacht. Dieses Kapitel befasst sich mit der Planung von Usability Engineering in Projekten.



Ziele erreichen

Bei der Optimierung der Benutzbarkeit von Systemen oder Produkten können unterschiedliche Schwerpunkte gesetzt werden. Der Einsatz geeigneter Usability-Methoden hängt von den Zielen ab, die ein Projekt erreichen soll. Nicht selten wird in Projektaufträgen pauschal eine „hohe Usability“ gefordert. Damit Usability-Aktivitäten jedoch überhaupt eingeplant werden können, müssen zumindest abschätzbare, wenn nicht sogar messbare Ziele festgelegt werden. Im Folgenden finden Sie eine Auswahl möglicher Ziele:

- Die Arbeitsgeschwindigkeit der Benutzer mit dem neuen System maximieren
- Die Anzahl notwendiger Schritte zur Ausführung einer Funktion und die dafür benötigte Zeit minimieren
- Den Ausbildungs- und Lernaufwand minimieren
- Die Qualität des Arbeitsergebnisses steigern

- Die Anzahl Fehler der Benutzer minimieren
- Das Gefährdungspotenzial aufgrund falscher Interpretationen oder Fehlmanipulationen reduzieren
- Die Zufriedenheit der Benutzer mit dem Produkt und ihrer Arbeit erhöhen
- Die Akzeptanz eines neuen Produktes erhöhen

Ein Beispiel dazu: Will die Geschäftsleitung die Zeit reduzieren, die ein Kunde beim Musikhören in der Warteschlange eines Callcenters verbringt, kann ein in der Benutzung effizienteres System dazu beitragen. Je schneller die Mitarbeiter die Anrufe im Mittel abarbeiten, desto kürzer ist die Wartezeit der Anrufer. Die Reduktion der durchschnittlichen Bearbeitungszeit eines Kunden-Anrufs ist ein messbares Ziel, das verfolgt werden kann. Es bestimmt, welche Usability-Methoden mit wie viel Aufwand im Projekt einzuplanen sind.

Risiken kontrollieren

Ein weiterer wesentlicher Faktor für die Planung von Usability Engineering ist die Risikobetrachtung eines Projekts. Dabei gilt es, Risiken zu identifizieren und gezielt anzugehen. Vereinfacht lassen sich Projekt- und Produktrisiken unterscheiden:

Projektrisiken gefährden den erfolgreichen Abschluss eines Projekts. Sie sind oft technischer und projekt-politischer Natur. Zwei typische Projektrisiken, welche mit Usability Engineering angegangen werden können, sind schlechte Akzeptanz bei den Benutzern sowie zu viele, erst spät entdeckte Anforderungen.

Produktrisiken hingegen gefährden den erfolgreichen Einsatz eines Produktes. Usability-Aspekte spielen bei Produktrisiken oft eine entscheidende Rolle. Die Identifikation solcher Produktrisiken ist für die Einplanung von Usability-Maßnahmen im Projekt zentral. Dazu einige Beispiele:

- Systeme mit mangelnder Usability erfordern vom Benutzer mehr Aufmerksamkeit und lenken ihn von seiner eigentlichen Tätigkeit ab. Dieses Risiko mag beispielsweise für Büroanwendungen tolerierbar sein – für ein neuartiges Navigationsgerät im Auto muss es dagegen mit geeigneten Methoden minimiert werden.
- Fehleingaben können unangenehme Folgen haben. Bei einem System für den Börsenhandel können sie sogar fatale Auswirkungen haben, falls beispielsweise ein Händler ungewollt 10 000 000 statt

10000 Aktien verkauft. Solche Fehleingaben müssen unter allen Umständen vermieden werden.

- Eine unklare Benutzerschnittstelle stellt bei der Entwicklung eines Farbenmischgeräts ein anderes Risiko dar als bei einem Gerät für die Dosierung von Medikamenten. Entsprechend hoch sind die Anforderungen bezüglich Usability.

Denkanstoß: *Sind Sie zurzeit in ein Projekt involviert? Überlegen Sie sich, welche Ziele das Projekt erreichen will und welche Risiken bestehen. Zu welchen dieser Ziele und gegen welche Risiken kann Usability Engineering einen wesentlichen Beitrag liefern?*

Rahmenbedingungen

Neben den zu erreichenden Zielen und den zu beherrschenden Risiken ist der Einsatz von Usability-Methoden von diversen Rahmenbedingungen abhängig. Diese Faktoren beeinflussen die Planung von Usability Engineering im Projekt maßgeblich. Beispielsweise lässt sich Contextual Inquiry nur durchführen, wenn das Projektteam tatsächlich mit Benutzern vor Ort sprechen kann. Im Folgenden ist eine Auswahl weiterer Einflussfaktoren aufgeführt:

- Wie verfügbar sind die Benutzer für Beobachtungen, Interviews und Workshops?
- Handelt es sich um interne oder externe Benutzer?
- Wie intensiv arbeiten die Benutzer mit dem Produkt und wie groß ist ihr Fachwissen?
- Findet die Entwicklung intern, extern oder off-shore statt?
- Muss auf Personen mit besonderen Merkmalen geachtet werden, z. B. Kinder, ältere Personen, Analphabeten, Personen mit Behinderungen?
- Wird spezielle Hardware oder besondere Software-Technologie vorgegeben, z. B. mehrere Bildschirme, Joysticks, Spracheingabe, Touchscreens, 3-D-Grafik?
- Kommt das System an einem besonderen Standort zum Einsatz?
- An welchem Punkt der Entwicklung steht das Projekt, wie weit sind die Projektziele, die Anforderungen, die Architektur und das User Interface bereits fortgeschritten und gefestigt?
- Welche Kenntnisse sind im Projektteam im Bereich Usability Engineering vorhanden? Müssen Berater zugezogen werden?

- Welche Firmenrichtlinien, wie Corporate-Design-Vorgaben und Vorgehensstandards, müssen berücksichtigt werden?
- Nach welchem Entwicklungsprozess wird vorgegangen?

Planungsbeispiele

Ziele, Risiken und Rahmenbedingungen bilden die Grundlage für die Planung. Aus diesen Eckpfeilern lässt sich ableiten:

- Welches die am besten geeigneten Usability-Methoden sind
 - Zu welchem Zeitpunkt diese zum Einsatz kommen
 - Mit wie viel Aufwand gerechnet werden muss
 - Welche Personen mit welchen Fähigkeiten eingesetzt werden müssen
- Die folgenden Abschnitte illustrieren dies anhand von drei prototypischen Projekten.

Ein internes Datenverarbeitungssystem

Fokus	Effizienz und Effektivität
Benutzer	Interne Mitarbeiter; ca. 700 Benutzer; ausgebildetes Fachpersonal, das intensiv mit dem System arbeitet
Usability-Ziele	Produktivitätssteigerung der Mitarbeiter; sehr gute Einpassung in erneuerte Prozesse; Arbeitsweise optimieren; effizientes User Interface; kurze Schulung; Training on the job
Projektrisiken	Schlechte Akzeptanz; viele Nachbesserungen
Produkttrisiken	Sinkende Effizienz der Mitarbeiter; hoher Aufwand für Schulung und Support
Projektstand	Das noch inoffizielle Projektteam erarbeitet den Projektantrag. Das Team umfasst später über 50 Personen, davon mehr als 20 Software-Entwickler.
Prozess	Phasenmodell mit Initialisierung, Geschäftsprozessmodellierung, Spezifikation, Realisierung, Test, Einführung und Betrieb

Folgende Usability-Methoden kommen zum Einsatz:

- **Contextual Inquiry:** in der Initialisierung Verbesserungspotenzial erkennen; während der Geschäftsprozessmodellierung und in der Spezifikationsphase Detailfragen klären.
- **Szenarien:** zu Beginn der Spezifikationsphase fachliche Abläufe mit der neuen Applikation entwerfen und validieren; in der Testphase als Basis für Testfälle und Usability-Testaufgaben einsetzen.
- **Storyboards:** in der Initialisierung eine gemeinsame Vision erarbeiten; während der Spezifikation Abläufe und Konzepte visualisieren und Feedback einholen.

- Use Cases: in der Spezifikation den Systemumfang festlegen und Anforderungen dokumentieren.
- UI Prototyping: während der Spezifikation Funktionalität und Informationsgehalt mit den Fachabteilungen erarbeiten.
- Usability Walkthroughs: während der Spezifikation die User-Interface-Prototypen evaluieren.
- Projekt-Styleguide: während der Realisierung die Entwickler unterstützen.
- Usability Testing: als Teil der Abnahme vor der Einführung.
- Fragebögen: im Betrieb die Erfüllung der Usability-Ziele auch nach mehreren Wartungsupdates prüfen.

Beim geschilderten Vorgehen sollte ein Projektleiter auf die folgenden Punkte achten:

- Datenverarbeitungssysteme zeichnen sich oft durch viele Spezialfälle in der Anwendung aus. Diese werden typischerweise erst durch die Betrachtung konkreter Beispiele sichtbar. Jeder nicht durch das System berücksichtigte Spezialfall bringt zusätzlichen Arbeitsaufwand in der Benutzung. So besteht das Risiko, dass die Benutzer den Standardfall effizient behandeln, jedoch keine Unterstützung für die Besonderheiten der realen Arbeitswelt erhalten und damit Produktivität und Qualität sinken.
- Werden interne Benutzer eingebunden, dann sollte der Projektleiter regelmäßig über die erzielten Fortschritte informieren.

Ein neues digitales Wunderwerk für den Consumer-Markt

Fokus	Schlichtheit und Attraktivität
Benutzer	Konsumenten: verschiedene Altersgruppen; technisch Interessierte sowie absolute Anfänger; aus sämtlichen Kulturkreisen
Usability-Ziele	Sofort benutzbar ohne Training; gutes Design; überschaubare Menge an Funktionen
Produktisiken	Hohe Kosten für Support; überlastete Hotline; schlechte Verkaufszahlen; Imageverlust
Projektstand	Das Produktmanagement hat ein Set von Features, Rahmenbedingungen und die angestrebte Zielgruppe definiert. Das entstandene Lastenheft ist Vorgabe für das Projektteam.
Prozess	Marktanalyse, Konzept, Funktionsmuster, Serienreifmachung, Produktion, Markteinführung

Das digitale Wunderwerk, beispielsweise ein Harddisk-Videorekorder, hat einige Eigenschaften, die eine deutlich andere Planung als für das interne Datenverarbeitungssystem erfordern. Insbesondere die

heterogene Benutzergruppe stellt eine besondere Herausforderung dar. Das digitale Wunderwerk ist funktional einfacher als ein Datenverarbeitungssystem, zeichnet sich aber durch spezialisierte Hardware aus. Folgende Methoden werden eingesetzt:

- **Contextual Inquiry:** in der Marktanalyse als Ergänzung zur Marktforschung.
- **Personas und Szenarien:** in der Marktanalyse und Konzeptphase Benutzer und Abläufe modellieren und dokumentieren; Fokussierung auf das Zielpublikum.
- **UI Prototyping, Usability Walkthroughs:** in der Konzeptphase die Benutzerschnittstelle erstellen und optimieren; wöchentliche Iterationen.
- **Use Cases:** während der Konzeptphase die funktionalen Anforderungen festhalten.
- **Usability Testing:** Testen der Prototypen am Ende der jeweiligen Phasen; Testen des Produktes vor Markteinführung.

Beim geschilderten Vorgehen sollte ein Projektleiter auf die folgenden Punkte achten:

- Im Gegensatz zum internen Informationssystem müssen die für Usability Engineering bedeutsamen Informationen aus vielen Quellen zusammengetragen werden. Beispielsweise können relevante kulturelle Aspekte wie Modeströmungen, Vorlieben und mehr mit Methoden der Marktforschung erarbeitet werden. Weitere wichtige Informationsquellen sind die lokalen Vertriebsgesellschaften und der Kundendienst. Auch Contextual Inquiry ist für ausgewählte Fragestellungen zu Benutzung und Umfeld hilfreich.
- Benutzer erfahren ein solches Produkt als Ganzes. Software und Hardware müssen zusammenpassen. Das Gerät sollte von Beginn an als Ganzes dargestellt und evaluiert werden.

Wenn Leben auf dem Spiel steht

Fokus	Sicherheit
Benutzer	Personen mit fachlicher Ausbildung, ausgiebigem Training
Usability-Ziele	Fehlerrate bei Benutzern minimieren; Qualität des erreichten Resultates maximieren; Nachlässigkeiten und Umgehungen verhindern; effiziente Bedienung
Produkttrisiken	Unfälle mit schwerwiegenden Auswirkungen auf Mensch, Umwelt und Sachwerte
Prozess	Analyse, Konzept, Entwicklung, Testphase, Produktion

Produkte mit hohen Anforderungen an die Sicherheit finden sich in vielen Bereichen, beispielsweise in der Medizin, in der Automobilbranche, in der Flugsicherung, bei industriellen Anlagen und Weiteren. Fehlbedienungen, Nachlässigkeiten und Umgehung von Sicherheitsmaßnahmen sind Ursachen für Unfälle, die üblicherweise unter die Rubrik „menschliches Versagen“ fallen. Eine auf Sicherheit optimierte Benutzerschnittstelle verringert solche Unfälle, manchmal auch auf Kosten von Bequemlichkeit und Effizienz. Zum Beispiel werden Behälter für Medikamente heute so konstruiert, dass sie durch kleine Kinder nur schwer geöffnet werden können. Genauso werden bei kritischen Eingabefeldern keine Standardwerte vorgegeben, damit ein unaufmerksamer Benutzer das Feld nicht einfach überspringen kann. Folgende Methoden werden typischerweise eingesetzt:

- **Contextual Inquiry:** in der Analysephase die Aufgaben, die eintrainierten Verhaltensweisen sowie die Gestaltung bestehender Systeme im Detail verstehen und dokumentieren.
- **UI Prototyping:** in der Konzeptphase eine zu den fachlichen Abläufen passende Benutzerschnittstelle erarbeiten.
- **Usability Walkthroughs, Expertenreviews und Usability Testing:** in der Konzeptphase und zu Beginn der Testphase das System mit Benutzern überprüfen.
- **Usability Testing:** in der Testphase das System im Usability Lab validieren und die Qualität der mit dem System geleisteten Arbeit prüfen; im Feldtest Vorfälle bezüglich ungenügender Usability analysieren.

Beim geschilderten Vorgehen sollte ein Projektleiter auf die folgenden Punkte achten:

- Es ist eine 100%-Einstellung gefordert: Bei den meisten Optimierungen kann bereits mit wenig Aufwand eine messbare Verbesserung erzielt werden. Dies genügt bei kritischen Risiken nicht: Bereits ein falscher Standardwert oder eine falsche Bezeichnung eines Knopfes können die Eintretenswahrscheinlichkeit eines Risikos massiv erhöhen. Jedes Detail ist wichtig!
- Es ist nur 99 % Sicherheit erreichbar. Kein Produktrisiko kann vollständig verhindert werden.
- Usability Engineering allein genügt nicht. Risikomanagement muss bei hohen Risiken auf breiter Front ansetzen, z. B. mittels Notfallszenarien, Ausbildung und Zertifizierung der Benutzer, Analyse von Vorfällen und weiteren Maßnahmen.

- Im sicherheitskritischen Umfeld sind oft zusätzliche Usability-Normen zu berücksichtigen. In der Medizinaltechnik sind beispielsweise die Normen [IEC 60601-1-6] und [IEC 62366] relevant.

Einsatz von Benutzern

Für die Durchführung von Usability-Methoden werden oft gut qualifizierte Benutzer benötigt. Dies ergibt für ein Projektteam zusätzliche Aufgaben.

Benutzer rekrutieren

Das Rekrutieren von Benutzern kann ein schwieriges Unterfangen sein. In vielen Situationen muss ein Projektteam selbstständig Benutzer anwerben. Dabei sollte auf die folgenden Punkte geachtet werden:

- Überlegen Sie sich vor einer Untersuchung, welche Eigenschaften die eingeladenen Benutzer haben sollten, und prüfen Sie, ob diese erfüllt werden.
- Achten Sie auf ein breites Spektrum an Eigenschaften und Fähigkeiten der eingeladenen Benutzer.
- Besprechen Sie den Einsatz von internen Benutzern mit ihren Vorgesetzten.
- Halten Sie bei externen Benutzern mit Marketing und Verkauf Rücksprache, und achten Sie auf die notwendige Geheimhaltung.

Im Verlauf eines größeren Projekts wird ein Projektteam in der Regel eine ganze Reihe von Benutzern einbeziehen. Der Aufbau einer Benutzerdatenbank kann die Rekrutierung mit der Zeit stark vereinfachen.

Eine umfassende Darstellung zum Thema Rekrutierung von Benutzern und deren Einbezug in Usability-Aktivitäten gibt das Buch [Courage & Baxter 05].

Benutzer entlohnen

Eine angemessene Bezahlung der Benutzer ist angebracht. Bei externen Personen werden die Leistungen deshalb meist mit Geld, Gutscheinen oder Geschenken vergütet. Für interne Benutzer ist die Wertschätzung ihrer Mitarbeit oft die beste Abgeltung. Dies bedeutet, dass das Projektteam die Benutzer nutzbringend einsetzt, Anregungen und Kritik aufnimmt und darauf reagiert. Es ist die Aufgabe des

Projektleiters, die beteiligten Benutzer regelmäßig über den erreichten Fortschritt zu informieren.

Anonymität und Vertraulichkeit

Die Äußerungen von Benutzern müssen anonym und vertraulich gehandhabt werden. In einem Interview wird ein Analyst auch einmal negative Aussagen zu hören bekommen. Genauso wird er vielleicht Fehler und Verstöße beobachten. Die folgenden Punkte bezüglich des Datenschutzes müssen Sie beachten:

- Dem Teilnehmer einer Untersuchung sollte klar sein, wie die gesammelten Daten aufbewahrt werden und wie Anonymität und Vertraulichkeit eingehalten werden.
- Aussagen, die ein Projektteam aufbewahrt, sollten anonymisiert werden.
- Für Video- oder Audioaufnahmen ist das Einverständnis der Benutzer einzuholen. Klären Sie ab, ob Sie die Aufnahmen aufbewahren dürfen.
- Beachten Sie die Datenschutzgesetze. Diese regeln, wie mit personenbezogenen Daten umgegangen werden muss.

Benutzer einbeziehen ist Projektmarketing

Jeder Kontakt mit Benutzern ist eine Möglichkeit, deren Bedenken und Ängste zu erfüllen und sie über die Chancen, die das Projekt bietet, zu informieren. Usability-Methoden sind deshalb ein ausgezeichnetes Mittel, um die Akzeptanz bei Benutzern zu erhöhen. Die Bedingung dazu ist, dass das Projektteam auf die Kritik und die Bedenken der Benutzer eingeht. Vergessen Sie nicht: Auch wenn Sie *nicht* mit Benutzern sprechen, kommunizieren Sie!

Strategische Usability

Ausgerüstet mit der Kenntnis der wichtigsten Usability-Methoden, mit einer guten Planung in der Tasche und den richtigen Leuten an Bord beginnt Ihr nächstes Projekt, in dem endlich ein wirklich benutzbares Produkt entstehen soll. Und dann wird alles ganz anders: Das Budget für Usability-Tätigkeiten wird gekürzt, in einem Management Board wird über neue Funktionalität abgestimmt, und Sie reden seit einer Woche an Wände – kurzum: Sie befinden sich in den organisatorischen Mühlen eines jeden (größeren) Unternehmens.

Die vorherigen Kapitel haben gezeigt, wie Usability-Methoden in ein Projekt für die Software- oder Produktentwicklung integriert werden können. Dieses Kapitel gibt Hinweise:

- Wie Usability unternehmensweit platziert werden kann.
 - Wie Sie es schaffen, dass die involvierten Stellen in einem benutzerorientierten Prozess zusammenspielen.
 - Welche strategischen Aspekte in Usability Engineering stecken.
- Dieses Kapitel dürfen Sie auch Ihrem Chef zeigen!

Usability unternehmensweit

Ein benutzerorientiertes Vorgehen in einem Unternehmen zu etablieren, bedeutet in der Regel, einige zementierte Ansichten aufzubrechen. Es mag auf den ersten Blick unkonventionell erscheinen, Kundenberater und Entwickler in einem Prototyping Workshop zusammenzubringen oder einen halben Tag am Arbeitsplatz eines Benutzers zu verbringen. Und dennoch wird dies zu Erkenntnissen führen, die im weiteren Verlauf des Projekts viel Zeit und Kosten sparen werden und das Ergebnis positiv beeinflussen. Um das in den vorigen Kapiteln beschriebene Vorgehen in einem Unternehmen einzuführen und die Organisation entsprechend weiterzubringen, ist eine gute Planung nötig.

Usability Engineering als Feedbackschleife

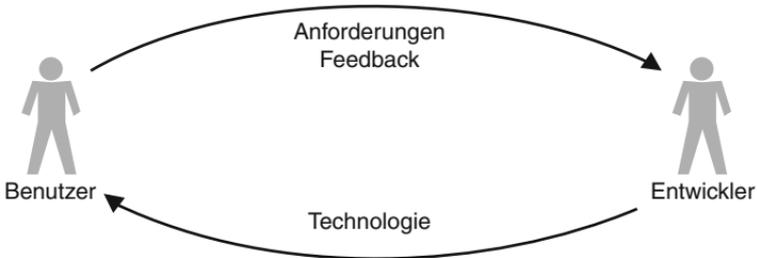
Usability Engineering hat zum Ziel, dass optimal auf die Benutzer zugeschnittene Lösungen erstellt werden. Dazu müssen

- Informationen über die Benutzer, deren Arbeitsabläufe, Bedürfnisse, Anforderungen, Aufgaben und Umgebung systematisch in die Software- oder Produktentwicklung einfließen und

Aufbau eines benutzerorientierten Prozesses

- die technischen Möglichkeiten, Grenzen und Rahmenbedingungen in einer verständlichen Form an die Benutzer zurückfließen.

Aus organisatorischer Sicht heißt dies nichts anderes, als eine Feedbackschleife zwischen Benutzern und Entwicklungseinheiten aufzubauen. Usability-Methoden unterstützen diesen Prozess, der in der folgenden Abbildung verdeutlicht wird:



Wer Usability Engineering in Organisationen auf diese Weise versteht, löst sich von der isolierten Betrachtung des Datensammelns, User-Interface-Gestaltens oder Usability-Testens. An die Stelle von Kritik, die dem Entwickler in Form des neusten Testberichts aus dem Usability Lab vor die Nase gehalten wird, tritt eine konstruktive Haltung: Usability Engineering bringt jene Stellen zusammen, die für die Entwicklung von benutzbaren Lösungen unverzichtbar sind: Benutzer und Entwickler.

Nun gilt es, die notwendigen Rahmenbedingungen zu schaffen, um dieses Ziel umzusetzen. Die folgenden Abschnitte zeigen drei praxiserprobte Ansätze, um Usability Engineering unternehmensweit zu platzieren: Den Aufbau eines benutzerorientierten Prozesses, die Einführung von Usability Standards und die Institutionalisierung von Usability im Unternehmen.

Aufbau eines benutzerorientierten Prozesses

Usability Engineering bewegt sich organisatorisch an den Schnittstellen zwischen Geschäftseinheiten und Entwicklung; in der Praxis wird häufig vom „Business“ und der „Technik“ gesprochen. Die Anforderungen der Benutzer sollen erhoben werden und unter Berücksichtigung der technischen Möglichkeiten und Grenzen in die Entwicklung einfließen. Um nachhaltig für benutzbare Lösungen zu sorgen, muss

ein Unternehmen einen Entwicklungsprozess etablieren, welcher die Feedbackschleife zwischen Business und Technik zum richtigen Zeitpunkt ermöglicht.

Um den benutzerorientierten Aktivitäten das notwendige Gewicht zu verleihen, ist es wichtig, entsprechende Rollen und Lieferergebnisse in diesen Prozess aufzunehmen und die notwendigen Ressourcen (Zeit und Budget) bereitzustellen. In einem größeren Unternehmen wurde beispielsweise die Projektrolle „Usability Engineer“ definiert. Deren Verantwortung umfasste die Begleitung des Entwicklungsteams bei der Durchführung von Usability-Methoden und die Erstellung von User-Interface-Konzepten als Teil der Spezifikation.

Es bringt dabei wenig, theoretisch korrekte, aber nicht akzeptierte Vorgaben einzuführen. Die benutzerorientierten Aktivitäten müssen in den bestehenden (gelebten) Entwicklungsprozess eingepasst werden. Die Berücksichtigung der folgenden drei Punkte erachten wir als essenziell:

- *Business-Analyse und Requirements Engineering*: Benutzerorientierte Methoden müssen dort ansetzen, wo Anforderungen von der Business- oder Kundenseite erarbeitet, analysiert und in Spezifikationen übersetzt werden. Dort werden die Weichen für eine benutzbare Lösung gestellt. Später im Prozess kann nur noch korrigiert werden.
- *Iteratives Vorgehen*: Es sollte ein iterativer Prozess aufgesetzt werden, in welchem Anforderungen und Spezifikationen visualisiert, mit Benutzern und Geschäftseinheiten überprüft und falls notwendig angepasst werden können. Wenn formale Freigaben erfolgen oder Fachstellen Spezifikationen verabschieden müssen, dann sollten diese iterativen Tätigkeiten möglichst vorher stattfinden.
- *Gemeinsame konkrete Sprache*: Um eine benutzbare Lösung zu erreichen, müssen Benutzer, Business-Einheiten, Fachstellen und Entwickler ein gemeinsames Verständnis entwickeln. Dies ist allein mit formalen, abstrakten Beschreibungen nicht möglich. Beispielerorientierte Methoden wie Szenarien, Storyboards und User-Interface-Prototypen fungieren als verbindende Aktivitäten in der Produktentwicklung. Das User Interface ist die gemeinsame Sprache der verschiedenen Stellen und erlaubt eine effiziente Zusammenarbeit.

Checkliste für die Anpassung Ihres Entwicklungsprozesses

- *Methoden*: Welche Usability-Methoden unterstützen in welchen Phasen der Projekte am meisten?

- *Lieferergebnisse:* Usability Engineering verwendet bestimmte Ergebnisse und Modelle, um Lösungen zu erarbeiten und Resultate festzuhalten. Welche benötigen Sie? Sollen diese mit bestehenden Lieferergebnissen zusammengeführt oder zusätzlich erstellt werden?
- *Werkzeuge:* Beim Erstellen von informellen, bildlichen und auf Prototypen basierenden Ergebnissen können geeignete Werkzeuge helfen (siehe auch Abschnitt „UI Prototyping Tools und -Komponenten“, S. 90). Wie können Sie von bestehenden Tools profitieren, welche Werkzeuge müssen Sie neu einführen?
- *Feedbackschleife:* Usability Engineering analysiert, interpretiert, erarbeitet Lösungsvorschläge und überprüft diese mit Benutzern in einem iterativen Prozess. Wo lassen sich solche Feedbackschleifen in Ihrem Unternehmen integrieren?
- *Rollen und Tätigkeiten:* Ein benutzerorientierter Prozess führt zu neuen Tätigkeiten und Projektrollen. Wer in Ihrem Unternehmen kann diese wahrnehmen? Wie funktioniert die Zusammenarbeit im Team?
- *Prinzipien:* Usability Engineering baut auf bestimmten Prinzipien auf (vergleiche Kapitel „Rückblick“, S. 117). Auf welchen Prinzipien gründet der Entwicklungsprozess in Ihrer Firma? Könnte dies zu Widersprüchen führen?

Usability-Standardisierung

Unternehmens-Styleguides

Im Kapitel Usability-Methoden wurde gezeigt, wie User Interface Storyboards und UI-Prototypen verwendet werden können, um die Anforderungen zwischen Business und Technik auszutauschen. Sie haben den Einsatz von Styleguides auch bereits als wichtiges Mittel für die Entwicklung einheitlicher Benutzerschnittstellen kennengelernt (siehe Abschnitt „Usability Guidelines und Styleguides“, S. 54). In allen Fällen übernimmt das User Interface die Funktion einer gemeinsamen Sprache zwischen den Beteiligten. Aus organisatorischer Sicht verkörpern Unternehmens-Styleguides gewissermaßen Wörterbuch und Grammatik dieser Sprache, indem sie User-Interface-Elemente mit Bezeichnungen und Anwendungsregeln festlegen. Durch eine solche Standardisierung können Konsistenz und Qualität der erstellten User Interfaces in einem Unternehmen nachhaltig verbessert werden.

Die Entwicklung eines Unternehmens-Styleguides ist besonders lohnenswert, wenn viele ähnliche Anwendungen realisiert werden, die sich bezüglich Zielgruppe, Verwendungszweck und eingesetzter Technologie gleichen. Dann lassen sich standardisierte User-Interface-Elemente definieren, die in verschiedenen Projekten verwendet werden können. Bei stark unterschiedlichen Anwendungen oder schnell wechselnden Technologien ist der Aufwand für die Ausarbeitung und Aktualisierung eines Unternehmens-Styleguides dagegen sehr hoch.

Der Aufwand für den Einigungsprozess bei der Erarbeitung eines Unternehmens-Styleguides sollte nicht unterschätzt werden. Oft müssen Anforderungen aus vielen unterschiedlichen Projekten abgestimmt und auf einen gemeinsamen Nenner gebracht werden. Zudem gilt es, zahlreiche Aspekte wie Ergonomie, technische Umsetzbarkeit, Corporate Design, Ästhetik und einige mehr zu berücksichtigen.

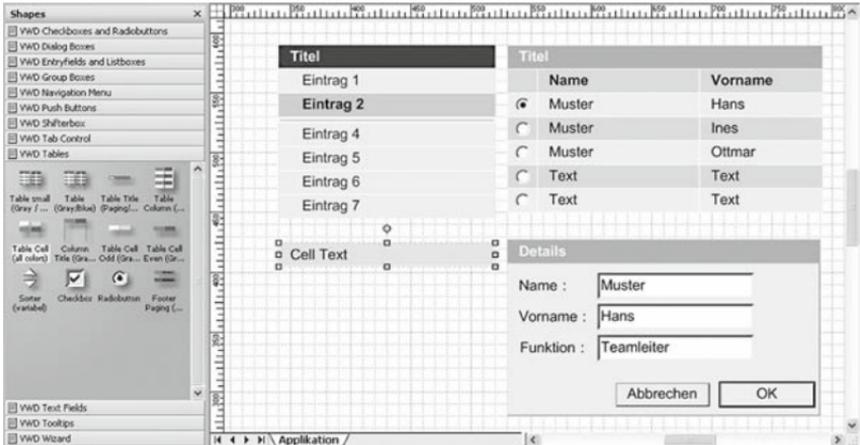
Unternehmens-Styleguides sollten keinesfalls als bloßes Regelwerk die strikte (und manchmal blinde) Einhaltung von Vorgaben erzwingen. Ein guter Styleguide trägt dazu bei, dass Anwendungen schneller und einfacher erstellt werden können. Styleguides entfalten in einem Unternehmen nur dann ihre Stärke, wenn sie mit guten Beispielen und Hilfsmitteln ein gemeinsames Verständnis und das sinnvolle Anwenden der Regeln fördern. Der folgende Abschnitt beschreibt zwei solcher Hilfsmittel.

UI Prototyping Tools und -Komponenten

Es ist immer wieder verblüffend, wie Personen auf abstrakter Ebene aneinander vorbei diskutieren und welche Missverständnisse sich klären, sobald jemand eine erste Skizze einer Benutzerschnittstelle auf ein Blatt Papier kritzelt. Was im Kleinen funktioniert, nützt auch im Rahmen einer größeren Organisation: Die Visualisierung von Anforderungen durch User-Interface-Entwürfe hilft, Missverständnisse zu vermeiden. Dieser Prozess kann mit folgenden Hilfsmitteln unterstützt werden:

Prototyping Tools stellen die in einem Unternehmens-Styleguide definierten User-Interface-Elemente zur Erstellung erster Entwürfe zur Verfügung (siehe auch Abschnitt „UI Prototyping“, S. 40). Je einfacher diese Werkzeuge sind, desto eher sind auch technisch unbedarfte Personen bereit, ihre Ideen und Vorstellungen auf Papier zu bringen. Elementsammlungen für einfache Zeichnungswerkzeuge oder bereits vorhandene Office-Applikationen, mit denen Benutzeroberflächen baukastenartig zusammengestellt werden können, haben sich dafür

bewährt. Die folgende Abbildung zeigt ein Werkzeug auf Basis von Microsoft Visio® zur Erstellung von einfachen User-Interface-Prototypen. Die vordefinierten GUI-Elemente entsprechen den Vorgaben eines Unternehmens-Styleguides.



Auf der anderen Seite können die definierten GUI-Elemente in Form von ausprogrammierten Komponenten für die Entwicklung zur Verfügung gestellt werden. Solche Komponenten erleichtern den Entwicklern die Einhaltung der Styleguide-Vorgaben und erhöhen die Qualität der erstellten User Interfaces. Selbstverständlich muss der Aufwand hierfür wie bei jeder Engineering-Tätigkeit sorgfältig abgewogen werden.

Die Kombination von Styleguides, Prototyping Tools und GUI-Komponenten übt eine Schlüsselfunktion für einen benutzerorientierten Prozess aus:

- UI-Elemente mit einer konsistenten Bezeichnung und Verwendung erlauben einen durchgängigen Prozess von der Spezifikation bis zur Implementierung und vereinfachen die Kommunikation und das Verständnis zwischen Benutzern, Geschäftseinheiten und Entwicklern.
- Die technischen Möglichkeiten sind schon in der Spezifikation sichtbar und werden auch von Nicht-Technikern verstanden. Geschäftseinheiten sind damit in der Lage, ihre Anforderungen innerhalb der technischen Rahmenbedingungen zu formulieren und bereits mit ersten Entwürfen zu visualisieren.

Usability-Institutionalisierung

Ein eigenes Usability Team

Während kleinere Organisationen in der Regel Usability-Dienstleistungen von spezialisierten Anbietern in Anspruch nehmen, kann es sich für größere Unternehmen mit einer Vielzahl von Entwicklungsprojekten lohnen, ein eigenes Usability Team aufzubauen. Usability Know-how im Hause zu haben, bringt eine Reihe von Vorteilen mit sich:

- Experten unterstützen langfristig die Etablierung von Usability-Methoden im Entwicklungsprozess und können mit internen Marketingmaßnahmen und Schulungen das Bewusstsein für das Thema nachhaltig stärken.
- Das Team baut das für Usability Engineering erforderliche Bereichs- und Branchenwissen auf.
- Interne Usability-Spezialisten können bereits in frühen Phasen eines Projekts mitwirken, wenn unter Umständen noch kein externes Budget verfügbar ist.

Die Nachteile sind:

- Die für Usability Engineering positive unvoreingenommene externe Sicht geht mit der Zeit verloren; es stellt sich auch beim besten Usability-Spezialisten eine gewisse „Betriebsblindheit“ ein.
- Die Akzeptanz ist oftmals geringer als gegenüber externen Spezialisten. Intern wird das Vertreten der Benutzersicht oft als Kritik empfunden, während man von Externen gerne die ungeschminkte Wahrheit sucht.

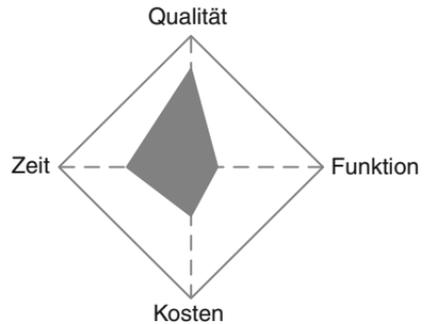
Chefsache: Usability als Teil der Geschäftsstrategie

Kundenzufriedenheit, einfach zu benutzende Produkte und effiziente Prozesse stehen meist ganz oben auf der strategischen Landkarte von Unternehmen. Immer öfter erscheint Usability auch als Begriff in Strategiepapieren und Visionsdokumenten. Was fehlt, ist indessen der Bezug zu der Planung und den Aktivitäten, die helfen sollen, diese strategischen Ziele zu erfüllen.

Usability ist ein Qualitätskriterium, das durch benutzerorientierte Methoden im Entwicklungsprozess erhöht werden kann. Die Geschäftsleitung muss definieren, mit welcher Priorität und mit welchen Ressourcen dieses Ziel verfolgt werden soll. Die folgende Abbildung stellt die Abhängigkeiten dar, die es dabei zu berücksichtigen gilt: Qualität, Funktionsumfang, Entwicklungszeit und Kosten stehen in einem Zielkonflikt.

Wie sieht es in Ihrem Unternehmen aus?

Es kann für einen Marktführer zum Beispiel wichtig sein, als erstes Unternehmen ein neues Produkt mit gegebenem Funktionsumfang und gerade noch akzeptabler Qualität zu lancieren und dafür hohe Kosten in Kauf zu nehmen. Es ist hingegen nicht möglich, alle vier angegebenen Aspekte gleichzeitig zu optimieren. Wer eine Applikation mit hohem Funktionsumfang in kurzer Zeit durchboxen will, darf sich nicht wundern, wenn die Usability für die späteren Anwender auf der Strecke bleibt.



Die Entscheidung für Qualität ist eine wesentliche Voraussetzung, dass Usability-Aktivitäten im Unternehmen mit der erforderlichen Priorität durchgeführt werden können. Dann kann Usability Engineering zum Unternehmenserfolg beitragen, Kunden nachhaltig begeistern und Mitarbeitern wirklich effiziente Anwendungen liefern.

Denkanstoß: *Spiegelt die Vision Ihres Unternehmens die originären Bedürfnisse Ihrer Kunden wider? Leiten sich Ihre Mission Statements aus den grundlegenden Problemen der Benutzer ab oder stehen vielmehr technologiegetriebene Ansätze dahinter?*

Google publiziert auf seiner Website (www.google.com) die Firmenphilosophie in zehn Punkten. Der erste Punkt ist: „Focus on the user and all else will follow.“ Wie ist das Leitbild ihrer Firma?

Wie sieht es in Ihrem Unternehmen aus?

Überlegen Sie sich die folgenden Punkte für das Unternehmen, in dem Sie arbeiten. Betrachten Sie das letzte Projekt, an dem Sie beteiligt waren.

- Auf welchem Weg gelangen heute die Anforderungen der Benutzer in die Entwicklung?
- Welches sind die beteiligten Stellen, Prozesse und Instrumente?
- Wie erfolgt das Feedback der Benutzer?
- Wie zufrieden waren die Benutzer mit dem Resultat? Ist dies überhaupt bekannt?

Beurteilen Sie Schwachstellen und Stärken dieses Vorgehens. Wo können Sie mit gezielter Institutionalisierung von Usability Engineering eine Verbesserung erreichen?

Denkanstoß: Betrachten Sie die folgenden realen Beispiele anhand der aufgeführten Punkte. Gibt es typische Muster?

Beispiel 1

In einer Großbank wird eine spezialisierte Applikation für Kundenberater von einer internen Software-Entwicklungsabteilung erstellt. Die Anforderungen der Geschäftseinheiten werden von Benutzervertretern eingebracht und von einer speziellen Einheit von Business-Analysten gesammelt. Sie erstellen Spezifikationen und geben diese in Form von umfangreichen Dokumenten an die Entwickler weiter. Der Entwicklungsprozess des Unternehmens sieht Sign-offs vor, in denen die Spezifikationen von den verschiedenen Fachstellen geprüft und freigegeben werden, bevor schließlich implementiert wird. Zwischen den Entwicklungs- und Business-Einheiten gibt es keine direkte Kommunikation. Die Kundenberater werden erstmalig bei Einführung mit der neuen Anwendung konfrontiert. Die Akzeptanz für die neue Lösung ist denkbar schlecht, die Kundenberater arbeiten mehrheitlich mit dem alten System weiter.

Beispiel 2

Im Bereich der öffentlichen Verwaltung ist eine zentrale Einheit für die Definition und Einführung von neuen IT-Lösungen für die rund 40 Geschäftsstellen mit Kundenkontakt zuständig. Um die Anforderungen zu sammeln, finden regelmäßige Boardmeetings der Geschäftsstellenleiter mit dem Leiter der zentralen Einheit statt. Die Anforderungen werden dokumentiert und zum größten Teil an externe IT-Dienstleistungsanbieter in Auftrag gegeben. Bei der zentralen Einheit arbeiten zwei bis drei ehemalige Geschäftsstellen-Mitarbeiter, die das Wissen über die Geschäftsfälle einbringen und die zudem den telefonischen Support für die Geschäftsstellen übernehmen. Das Feedback aus den Geschäftsstellen ist unterschiedlich, die eingeführten Lösungen sind vielfach zu komplex.

Beispiel 3

In einem Unternehmen der Maschinenbauindustrie ist die Forschungs- und Entwicklungsabteilung zuständig für die Weiterentwicklung der Maschinen. Die Kundschaft ist international und umfasst unterschiedliche Kulturen. Der Produktmanager formuliert zusammen mit dem

Marketing die Anforderungen an die neue Produktlinie und hält diese in einem Lastenheft fest. Als Grundlage dienen das Feedback der Vertriebsgesellschaften und die Features der Konkurrenz. Im Entwicklungsprojekt werden Maschine, Steuerung und Bedienpanel voneinander getrennt spezifiziert und entwickelt. Erstes Feedback durch Kunden erhält das Projekt an Fachmessen und durch ausgewählte Verkäufer an Marketingveranstaltungen. Die Maschine integriert sich gut in den Produktionsprozess der Kunden, ist jedoch umständlich zu benutzen und für angelerntes Personal kaum verständlich.

Darauf sollten Sie achten

In vielen Fällen können Benutzer und Projektteam nicht direkt miteinander kommunizieren. Vielleicht gibt es noch gar keine Benutzer für ein neues Produkt, oder die Benutzer sind schwer greifbar. Oft erzeugen Entscheidungsträger aber Distanzen und Barrieren aus unternehmerischen Überlegungen mit entsprechenden Auswirkungen auf die Kommunikation:

- *Geografische Entfernung:* Die Software- oder Produktentwicklung aus Kostengründen an einen billigeren Standort auszulagern, erzeugt Distanz zum eigentlichen Geschäft und damit schwierigere Kommunikationswege.
- *Kulturelle Hürden:* Neben der geografischen Distanz kommen Sprachbarrieren und Kulturunterschiede hinzu.
- *Unternehmensgrenzen:* Technische Funktionen werden an Drittfirmen und deren Zulieferer vergeben. Im internationalen Geschäft wird zusätzlich über Vertriebspartner und Servicestellen gearbeitet. Prozesse, Management und Unternehmensziele sind unterschiedlich.
- *Organisatorische Entfernung:* Auch innerhalb eines Unternehmens sind Benutzer organisatorisch von der Entwicklung weit entfernt. Auch wenn Zwischenstellen wie Benutzervertreter, Business-Analysten, Vorgesetzte und Fachstellen wichtige koordinierende Funktionen übernehmen, muss der Informationsfluss dabei viele Stellen durchlaufen.
- *Methodische Lücken:* Business-Analysten halten Anforderungen in abstrakten Spezifikationen fest, Marktforschungseinheiten verpacken Erkenntnisse über Zielgruppen in Statistiken und Vorgesetzte fassen Benutzerbedürfnisse zusammen. So notwendig die Konsolidierung der Benutzeranforderungen auch ist – falsche Methoden verhindern, dass wichtige Informationen über die Benutzer den Weg zu den Entwicklern finden.

That's life: Beispiele aus der Praxis

*Die einen, so scheint mir, haben viele Werkzeuge und wenig Ideen;
die anderen haben viele Ideen und gar keine Werkzeuge.*
(Denis Diderot)

Fallstudien

In diesem Abschnitt möchten wir Ihnen vier Projekte präsentieren, die den Einsatz der vorgestellten Usability-Methoden in der Software- und Produktentwicklung näher beleuchten. Vielleicht kommt Ihnen die eine oder andere Herausforderung bekannt vor?

Fallstudie 1: Usability und Requirements Engineering sorgen für gutes Klima

Im Rahmen der Neuentwicklung einer Berechnungs- und Auslegungssoftware für einen Hersteller von Luftbefeuchtungsanlagen wurden Usability- und Requirements-Engineering-Aufgaben so integriert, dass mit wenig Aufwand eine effiziente und einheitliche Anwendung entstand.

Steckbrief

Benutzer	Verkäufer der verschiedenen Ländervertretungen
Produkt	Software zur Auslegung von Luftbefeuchtungs-Anlagen mit Produktkatalog
Usability-Ziele	Verkaufsprozess effektiv und effizient gestalten, Konsistenz über alle Produktfamilien
Projektphase	Vision bis Realisierung
Methoden	Personas und Szenarien, Contextual Inquiry, UI Storyboards, UI Prototyping, Use Cases, Usability Walkthroughs

Ausgangslage

Luftbefeuchtungsanlagen kommen in den unterschiedlichsten Anwendungsgebieten zum Einsatz. In Bürogebäuden oder Einkaufszentren soll beispielsweise ein gesundes und angenehmes Luftklima entstehen, Krankenhäuser sind auf optimale Hygiene angewiesen, in Lebensmittellagern dürfen die Kartoffeln nicht verfaulen, während auf Baustellen die Aushärtung des Betons nur mit der notwendigen Luftfeuchtigkeit optimal verläuft. Dafür stehen unterschiedliche Technologien und Komponenten zur Verfügung, welche beispielsweise durch Verdampfung oder Einspritzung für die richtige Luftfeuchtigkeit sorgen.

Für eine erste Berechnung und Auslegung solcher Luftbefeuchtungsanlagen kommen Software-Anwendungen zum Einsatz, die den Verkäufer in der richtigen Auswahl und Dimensionierung der Komponenten für die Angebotserstellung unterstützen. Bei größeren Projekten sind auf Kundenseite meist externe Planer für die Konzipierung der Befeuchtungsanlagen verantwortlich.

Die bisher eingesetzten Berechnungsprogramme sind komplex und umständlich und zudem für jede Produktfamilie und Technologie unterschiedlich.

Herausforderungen

- Bei Projektstart ist noch nicht sicher, ob die neue Software ausschließlich für die Verkäufer oder alternativ auch für die externen Planer der Luftbefeuchtungsanlagen zum Einsatz kommen soll.
- Die neue Anwendung soll in den Ländervertretungen des Unternehmens weltweit eingesetzt werden. Die Ländervertretungen unterscheiden sich nicht nur in Sprache und regionalen Anforderungen, sondern auch stark bezüglich Größe, Verkaufsprozess und anzubindenden Systemen.
- Für verschiedene Produktfamilien stehen unterschiedliche Berechnungsprogramme und Produktkataloge zur Verfügung. Eine Herausforderung besteht darin, eine einheitliche Applikation zu erstellen, welche die gesamte Produktpalette des Unternehmens integriert, und dabei jegliche unnötige Komplexität für die Benutzer vermeidet.

Vorgehen

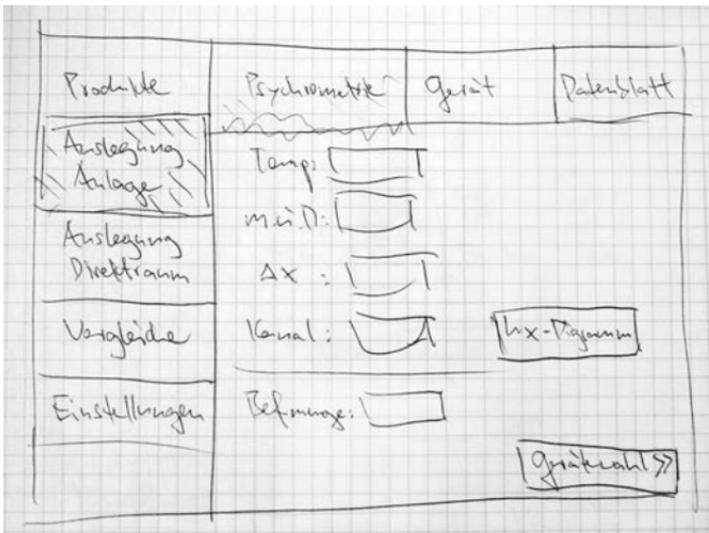
Die ersten Workshops mit dem Auftraggeber dienten der Analyse des Verkaufsprozesses im Hinblick auf die neue Lösung. Eine genauere Betrachtung der potenziellen Benutzergruppen ergab, dass die Bedürfnisse der Verkäufer und jene der Planer auf Kundenseite sich so stark unterschieden, dass sie nicht gleichzeitig mit der neuen Anwendung adressiert werden konnten. Als wichtiger Schritt musste deshalb die Zielgruppe eingegrenzt werden. Die Verantwortlichen entschieden, die geplante erste Version der Software ausschließlich für die Verkäufer zu entwickeln.

In einem Benutzerworkshop kamen Teilnehmer aus fünf verschiedenen Ländervertretungen zusammen, um realistische Personas und Anwendungsszenarien für den Einsatz der neuen Software zu erarbeiten. Die Szenarien lieferten eine wichtige Basis für die optimale

Unterstützung eines typischen Verkaufsablaufs und dienen der Eingrenzung und Fokussierung der Funktionalität für die Benutzer. Der Austausch zwischen den Teilnehmern erlaubte auch eine erste Abgrenzung der neuen Software gegenüber den in den Ländervertretungen eingesetzten Systemen, beispielsweise zur Erstellung und Ablage der schriftlichen Angebote oder für die Nachbestellung von Produkten und Komponenten bei der Muttergesellschaft.

Als nächster Schritt erfolgte ein Besuch in einer typischen Vertretung im Ausland. Die Verkäufer gaben Auskunft über ihren Arbeitsablauf vom Anruf des Kunden bis zur schriftlichen Angebotserstellung. Einige kritische Arbeitsschritte wurden vor Ort live beobachtet. Dabei konnten weitere wertvolle Informationen über die Arbeitsplätze, Prozesse und das Umfeld für den Einsatz der Software gesammelt werden. Die erarbeiteten Anforderungen wurden in einem Visionsdokument zusammengefasst.

Mit User Interface Storyboards holte das Team in kurzen Zyklen Feedback von den Fachvertretern ein. Die Storyboards zeigten die Abläufe für die Auswahl und Berechnung der Komponenten der Beefeuchtungsanlagen und gaben einen ersten Eindruck der vorgesehenen GUI-Konzepte der Applikation. Die folgenden Abbildungen zeigen verschiedene Ausführungen (unten: Handskizze, nächste Seite: Darstellung mit MS Visio®).



Auf Basis dieser Storyboards und der dokumentierten Anforderungen erstellte das Team einen lauffähigen GUI- und Architektur-Prototypen, der weitere Feedbackrunden am sichtbaren Objekt erlaubte. Parallel dazu wurden die funktionalen Abläufe mit Anwendungsfällen formal beschrieben und in mehreren Iterationen weiter detailliert.

Der GUI-Prototyp ermöglichte bald, in Usability Walkthroughs das Feedback der Benutzer einzuholen. Dazu bearbeiteten die Verkäufer mit dem Prototypen eine fiktive Kundenanfrage. Die Ergebnisse der Walkthroughs flossen in Form von weiteren Optimierungen in den Prototypen ein.

Die vollständige Realisierung der Applikation erfolgte schließlich auf Basis des Prototypen sowie der Spezifikation bestehend aus den Use-Case-Beschreibungen und zusätzlichen Anforderungen.

Nutzen und Fazit

- Mit der aktiven Eingrenzung der Zielgruppe in einer frühen Projektphase wurde die Produktvision wesentlich fokussierter. Dies

war ein erster wichtiger Schritt zur Vermeidung unnötiger Funktionalität und die Voraussetzung für eine gute Benutzbarkeit.

- Durch die frühe Einbeziehung von Benutzern aus verschiedenen Ländervertretungen und die gemeinsame Erarbeitung von Anwendungsszenarien konnten unterschiedliche Bedürfnisse sofort adressiert und ein gemeinsames Verständnis erzielt werden.
- Die Visualisierung der Arbeitsabläufe mit User Interface Storyboards und GUI-Prototypen erlaubte eine frühe Überprüfung der Anforderungen. Die Zusammenführung der verschiedenen Produktfamilien für die Luftbefeuchtung in eine einheitliche Lösung wurde sichtbar und konnte vom Produktmanagement und den Fachvertretern jederzeit begutachtet werden. Dank des iterativen Vorgehens konnten Lösungsvorschläge schnell auf die Bedürfnisse der Benutzer angepasst werden.
- Die parallele Erstellung des GUI-Prototypen und der Use Cases stellte sich als sehr effizientes Vorgehen heraus. Der GUI-Prototyp erlaubte es, die einzelnen Schritte der Anwendungsfälle am sichtbaren Objekt spielerisch nachzuvollziehen, während die Use Cases mit Alternativ- und Fehlerabläufen die Spezifikation vervollständigten. GUI-Abläufe und Use Cases befruchteten sich so gegenseitig.
- Die Vorgehensweise mit schlanken Methoden aus dem Usability- und Requirements Engineering führte mit wenig Aufwand zu guten Resultaten und insgesamt zu einer vergleichsweise kurzen Projektlaufzeit.

Fallstudie 2: Usability Engineering von A bis Z

Usability Engineering begleitet ein Projekt idealerweise in seiner ganzen Dauer, von den ersten Ideen bis zum Betrieb. Im Rahmen eines Projekts im Bahnumfeld wurden Usability-Methoden durchgängig in die Entwicklung integriert. Dies stellte sicher, dass die Anwender eine Software erhielten, welche den hohen Anforderungen ihrer täglichen Arbeit gerecht wurde.

Steckbrief

Benutzer	Operatives Personal mehrerer Bahnverkehrsunternehmen
Produkt	Individualsoftware für die Behandlung von Störungen im Bahnverkehr
Usability-Ziele	Schnell und mit wenigen Fehlern in einer anspruchsvollen Situation kommunizieren

Projektphase	Vision, Konzept, Realisierung in zwei Schritten
Methoden	Contextual Inquiry & Design, Storyboards, UI Prototyping, Usability Walkthroughs und Tests, Rollenspiele, Feedback-Fragebogen

Ausgangslage

Tritt im Bahnverkehr eine Störung auf, werden innerhalb weniger Minuten eine große Anzahl von Personen aktiv, um diese Störung zu beheben und die Auswirkungen auf die Kunden sowie die Züge zu minimieren. Dazu müssen viele Detailinformationen und Entscheidungen kommuniziert werden: Wie verläuft die Störung, was soll mit den betroffenen Zügen geschehen, wie können die Reisenden möglichst pünktlich an ihr Ziel gebracht werden und mehr. Bei größeren Ereignissen, beispielsweise wenn eine Strecke überhaupt nicht mehr befahren werden kann, werden vorbereitete Notfallkonzepte eingesetzt.

Die bisher übliche Kommunikation per Telefon ist redundant, fehlerbehaftet und kostet wertvolle Zeit, die für die eigentliche Ereignisbewältigung fehlt. Das neue System soll diese Informationen präzise und zuverlässiger zu den Benutzern bringen, sodass diese schnell und richtig handeln können.

Herausforderungen

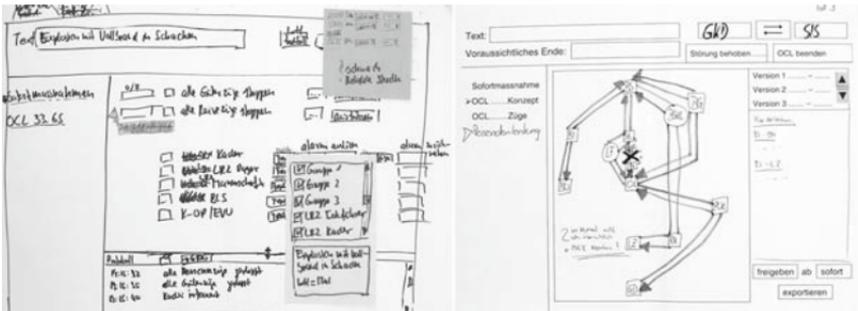
- Tritt eine Störung ein, sind die Anwender vollauf mit deren Bewältigung beschäftigt. Überflüssige Information ist möglichst zu vermeiden und der Aufwand für die Bedienung der Software zu minimieren.
- Mehrjährige Erfahrung und großes Fachwissen ist für die Störungsbehandlung notwendig – Wissen, das nur langjährige Mitarbeiter besitzen. Diese müssen deshalb in das Projekt eingebunden werden.
- Die Benutzer verlassen sich auf die Informationen im System. Falsche oder fehlende Angaben können kostspielige Folgen haben.

Vorgehen

Ein Team aus Anwendern und Fachexperten wurde von einem Usability Engineer unterstützt und legte mit Requirements- und Usability-Methoden die Basis für die Entwicklung. Anhand von Fallbeispielen skizzierte das Team Ideen und Ansätze mit Mock-ups auf Papier. Die

That's life: Beispiele aus der Praxis

folgende Abbildung zeigt gemeinsam erstellte Skizzen aus den Workshops (© Schweizerische Bundesbahnen, 2009).

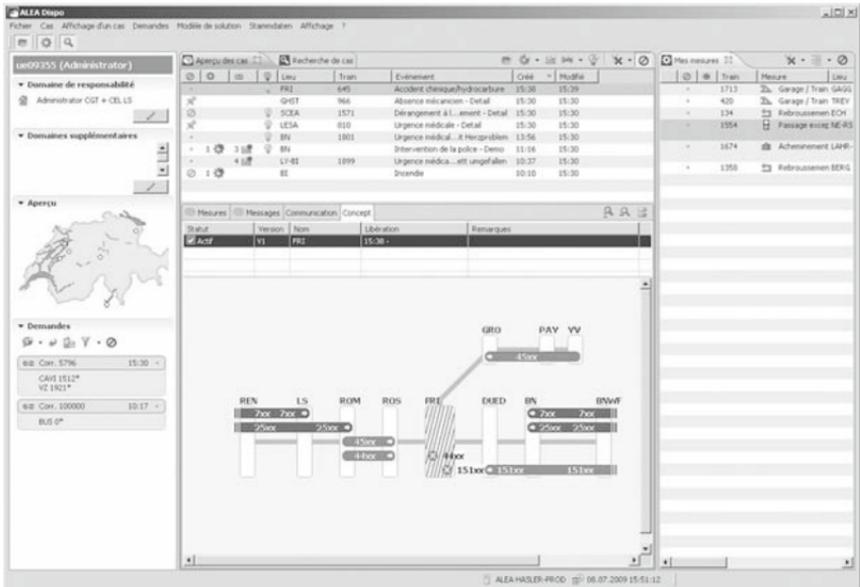


Ebenfalls analysierte das Team die Kommunikation in einer Störung. Dazu wurden Videokameras aufgestellt – mit der Bitte, diese in einem Störfall einzuschalten. Diese Aufnahmen und die anschließenden Interviews lieferten nicht nur wertvolle Detailkenntnisse, sondern schärften auch die Produktvision. Es zeigte sich beispielsweise, dass die Disponenten viele Informationen entgegennahmen und weitergaben – und wenig Zeit für die eigentliche Behandlung der Störung übrig blieb. Die Disponenten von der Rolle einer Informationszentrale zu entlasten, wurde zu einem wichtigen Fokus für die Software.

Die gefestigte Vision ermöglichte es nun, die vielen gesammelten Lösungsansätze zu konsolidieren und zu detaillieren. Das Team stellte dabei insbesondere ein Storyboard. Dieses visualisierte das neue System zum ersten Mal mit vielen wichtigen Details des User Interface, der angebundenen Systeme und im Zusammenhang mit der täglichen Arbeit. Nutzen und Bedenken konnten mit Mitarbeitern und Vorgesetzten diskutiert werden. Des Weiteren untersuchte das Projektteam die Auswirkungen auf die Prozesse. Rollenspiele halfen, die notwendigen Änderung an den Prozessen abzuschätzen und ein erstes Gefühl für die Umsetzbarkeit zu gewinnen.

Schließlich wurde ein interaktiver UI-Prototyp erstellt. Dieser zeigte einen großen Teil der geplanten Funktionen und war zusammen mit einer Anforderungsdokumentation der Ausgangspunkt für die Entwicklung. Aber erst, nachdem der UI-Prototyp in Usability Walkthroughs die Nützlichkeit des Systems hinreichend belegt hatte.

Für die nun startende Entwicklung wurde das Team um Software-Entwickler und einen Interaction Designer erweitert. Usability-Methoden wurden fortschreitend eingesetzt. Einerseits arbeitete das Team mit Mock-ups die Benutzerschnittstelle im Detail aus. Andererseits gaben punktuell eingesetzte Usability Tests Sicherheit und das nötige Wissen, um schwierigere Design-Probleme zu lösen. Beispielsweise wurde im mobilen Usability Lab die Geschwindigkeit der Tastatureingaben optimiert. Das iterative Vorgehen ermöglichte es dem Projektteam, noch vor Ablauf des Projekts eine erste Version in Betrieb zu nehmen. Das ungeschminkte Feedback der täglichen Arbeit konnte so in die Entwicklung einfließen. Die folgende Abbildung zeigt einen Screenshot der fertigen Software (© Schweizerische Bundesbahnen, 2009).



Nutzen und Fazit

- Experten der täglichen Arbeit, also künftige Benutzer, waren kontinuierlich im Projekt eingebunden. Das kritische Wissen der Störungsbehandlung war im Team verfügbar. Diese Teamzusammensetzung legte die Basis für die Nützlichkeit der Software.

- Usability-Methoden waren nicht allein das Arbeitsmittel eines Experten, sondern des Projektteams als Ganzem. Insbesondere auch die Anwender im Projektteam wurden in der Anwendung der Methoden geschult. So konnte jeder im Rahmen seiner Tätigkeit zur guten Benutzbarkeit beitragen.
- Dadurch, dass Benutzbarkeit systematisch und von Beginn an betrachtet wurde, besaßen realisierte Teile bereits eine hohe Usability. Während der Entwicklung selber konnten deshalb die im Verhältnis aufwändigen Usability-Tests sehr gezielt und damit sehr effizient eingesetzt werden: in erster Linie zur Optimierung besonders schwieriger Aspekte.
- Die komplexe, tägliche Arbeit war in einer künstlichen Situation, wie sie beispielsweise in einem Usability Lab erzeugt werden kann, nur ungenügend nachstellbar. Es war deshalb notwendig, bereits früh eine erste Version in Betrieb zu nehmen, um auf eventuelle Unzulänglichkeiten reagieren zu können.

Fallstudie 3: User Centered Innovation - Simulierte Realität

Wie kann Usability Engineering die frühen Projektphasen unterstützen, wenn weder Produkt noch Benutzer vorhanden sind? Diese Fallstudie beschreibt das benutzerorientierte Vorgehen für die Planung und Konzeption einer neuen Fernbedienung für Hörgeräte.

Steckbrief

Benutzer	Personen mit Hörschwächen
Produkt	Fernbedienung für Hörgeräte
Usability-Ziele	Dem Benutzer eine effiziente Einstellung auf die aktuelle Hörsituation ermöglichen; Komplexität vermeiden
Projektphase	Vorprojektphase, Analyse, Vision, Konzept, Requirements
Methoden	Interviews, Personas und Szenarien, Interaktionsdesign, UI Prototyping, Usability Testing

Ausgangslage

Die meisten Hörgeräte verfügen heutzutage über verschiedene Einstellmöglichkeiten, mit denen der Hörgeräteträger die jeweilige Hörsituation beeinflussen kann. Personen mit eingeschränktem Hörvermögen können beispielsweise die Lautstärke optimal auf ihre Bedürfnisse einstellen und zwischen Programmen für spezielle akustische Situationen wechseln. Die Bedienung findet je nach Modell ent-

weder direkt am Hörgerät statt oder alternativ über eine spezielle Fernbedienung, die der Benutzer bei sich trägt.

Aktuell angebotene Fernbedienungen bieten umfangreiche Funktionen, sind aber für viele Benutzer zu komplex. Eine zukünftige Produktpalette von Hörgeräten soll deshalb eine neuartige, stark vereinfachte Fernbedienung beinhalten. Diese neue Fernbedienung soll klein, leicht und sehr einfach in der Handhabung sein. Die Anforderungen der Träger von Hörgeräten an eine solche Fernbedienung sind aber noch weitgehend unbekannt.

Herausforderungen

- Das Projekt befindet sich in einer frühen Innovationsphase mit sehr vielen Freiheitsgraden für das zukünftige Produkt. Zielgruppen, Funktionsumfang, Größe und Form, technologische Lösungsvarianten, Tastenbelegung und viele andere Punkte sind noch offen und entsprechend viele unterschiedliche Meinungen und Vorstellungen über mögliche Lösungsansätze existieren im Unternehmen.
- Bei den potenziellen Benutzern, den Trägern von Hörgeräten, handelt es sich um eine schwer einzugrenzende Zielgruppe mit unterschiedlichen Bedürfnissen. Ein Großteil sind ältere Personen mit zunehmendem Hörverlust. Komplexe Funktionen und Bedienungsvorgänge sollten gerade für ältere Menschen vermieden werden. Eigenschaften und Verhaltensmuster, wie beispielsweise der Grad an Mobilität und Aktivitäten unterscheiden sich aber individuell stark. Dies führt zu abweichenden Anforderungen, was Häufigkeit und Art der Bedienung ihrer Hörgeräte anbelangt.
- Die Aufgabe des neuen Produkts ist das situative Anpassen des Hörgerätes für ein optimales Hören. Solche Hörsituationen sind nur schwer beobachtbar. Die für die Benutzer passende Einstellung ist für Außenstehende nicht objektiv nachvollziehbar.
- Die akustische Erlebniswelt und wichtige Konzepte bei der Bedienung von Hörgeräten lassen sich nur schwer in Worte fassen und beschreiben. Ein gemeinsames Verständnis für alle Projektbeteiligten ist deshalb umso wichtiger.

Vorgehen

In einer ersten Phase erfolgte eine Bestandsaufnahme im Unternehmen, um relevante Informationen über Hörgeräteträger und die Verwendung und Bedienung von Hörgeräten zusammenzutragen. Diese

Analyse bestand in erster Linie aus Gesprächen und Workshops mit Interessensvertretern und Spezialisten, beispielsweise Produktmanagement, Forschung und Entwicklung, Marketing und Marktforschung. Personen mit direktem Kundenkontakt konnten wertvolle Informationen über die Benutzerbedürfnisse beitragen, zum Beispiel Audiologen, die für die individuelle Anpassung des Hörgerätes für den Höreräteträger verantwortlich sind. Weitere wichtige Grundlagen waren Kundenfeedbacks zu bestehenden Produkten, Ergebnisse von Marktforschungs- und Benutzerstudien, Konkurrenzprodukte sowie Unternehmens- und Projektziele. In einem Visionsdokument wurden die gesammelten Resultate festgehalten.

In einem nächsten Schritt erstellte und priorisierte das Team Personas und Anwendungsszenarien für die neue Fernbedienung. Die Basis dazu lieferten Workshops sowie Interviews mit Höreräteträgern, in welchen die Bedürfnisse bei der Bedienung von Höreräten und dem Einsatz von Fernbedienungen ausgelotet wurden.

Für die weitere Definition von Form, Bedienelementen, Tastenbelegung und Funktionen der neuen Fernbedienung war es notwendig, verschiedene Varianten und Hardware-Prototypen mit Benutzern zu überprüfen. Das Ziel dabei war, mit Usability-Tests in kurzen Zyklen Resultate zu erzielen, problematische Lösungsvarianten auszuschließen und das Interaktionsdesign weiter zu verfeinern.

Zu diesem Zweck wurde eigens eine Prototyping-Plattform entwickelt, welche das Hörerlebnis für den Benutzer in beliebigen Umgebungen simuliert und einen realistischen Eindruck einer funktionstüchtigen Fernbedienung für Höreräte erzeugt. Mithilfe dieser Prototyping-Plattform konnten sowohl bestehende Anwendungsfälle für die Bedienung heutiger Höreräte überprüft werden, als auch Szenarien für zukünftige Hörerätegenerationen ausprobiert werden, die in dieser Form nicht existieren oder technisch noch nie umgesetzt wurden.

An den Usability-Tests nahmen Höreräteträger aus den möglichen Zielgruppen teil. Die Benutzer bekamen auf der Prototyping-Plattform realistische Hörsituationen präsentiert und interagierten mit verschiedenen Prototypen der Fernbedienung. Bereits die ersten Usability-Testserien konnten Hardware-Varianten ausschließen, weil sich Bedienelemente in der tatsächlichen Anwendung als vergleichsweise schlecht manipulierbar herausstellten, während andere bei den Benutzern sehr gut abschnitten. Die folgende Abbildung zeigt einige zu diesem Zweck erstellte Hardware-Prototypen der Fernbedienung

mit unterschiedlichen Tastenanordnungen, die in den Usability-Tests verwendet wurden.



Eine zentrale Fragestellung für die Entwicklung der neuen Fernbedienung war der Zusammenhang von Funktionsumfang und Manipulationsmöglichkeiten, welche das User Interface der Fernbedienung bieten sollte. Ein Ziel war beispielsweise, die Anzahl Tasten der Fernbedienung möglichst gering zu halten, und dabei für die Benutzer eine effiziente Bedienung der wichtigsten Einstellmöglichkeiten sicherzustellen. In den Usability-Tests zeigte sich, dass sowohl zu viele Tasten, als auch eine zu geringe Anzahl Tasten mit Mehrfachbelegung die Benutzer überforderte. Nach wenigen Iterationen konnte schließlich ein gut verständliches Interaktionskonzept mit einer optimalen Tastenbelegung gefunden werden.

Nutzen und Fazit

- Eine benutzerorientierte Methodik kann bereits in der Innovationsphase wichtige Aspekte aus Benutzersicht beitragen und damit neben technischen Rahmenbedingungen und strategischen Entscheidungen maßgebliche Anhaltspunkte für die Konzeption eines neuen Produktes liefern.
- Die Personas und Szenarien dienten den Projektbeteiligten zum Aufzeigen von impliziten Annahmen über Benutzer und Anwendungssituationen. Dies erlaubte Diskussionen und schaffte eine gemeinsame Vorstellung der Produktvision.
- Aufgrund der Benutzerbedürfnisse konnte bereits zu einem frühen Zeitpunkt eine Abschätzung der Wichtigkeit verschiedener Funktionen aus Benutzersicht vorgenommen werden. Das Vorgehen erlaubte eine Eingrenzung des geplanten Funktionsumfangs und damit eine Reduktion der Komplexität für die Benutzer.

- Die Prüfung von Varianten in iterativen Usability-Testserien diente der Risikominimierung, indem wenig erfolgreiche Lösungen bereits früh im Prozess ausgeschieden und erfolgreiche Design-Varianten weiter optimiert werden konnten.
- Die eigens für die Usability-Tests entwickelte Prototyping-Plattform ermöglichte eine schnelle Umsetzung von Varianten des Interaktionsdesigns. Es musste keine aufwändige Entwicklung in Kauf genommen werden, um lauffähige und testbare Prototypen auf der Hörgeräte-Zielplattform zu realisieren. Dies stellte sich als großer Vorteil heraus. Damit war es möglich, zu einem frühen Zeitpunkt ausgereifte Interaktions-Konzepte für die neue Fernbedienung zu erarbeiten und diese mit Benutzern zu überprüfen.

Fallstudie 4: Gezielter Einsatz, große Wirkung

Nicht in jedem Fall besteht die Möglichkeit, einen Experten für Usability Engineering über das ganze Projekt einzubinden. Wir möchten Ihnen deshalb hier ein Fallbeispiel zeigen, bei welchem ein kurzer Einsatz eine große Wirkung erzielte.

Steckbrief

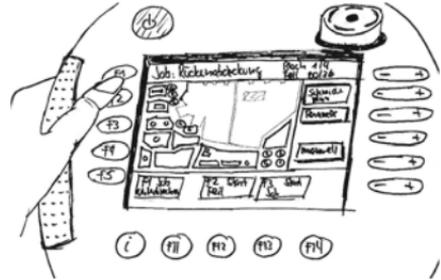
Benutzer	Operateure, Einrichter und Servicepersonal
Produkt	Bedienpanel einer Schneidemaschine für Bleche im industriellen Bereich
Usability-Ziele	Einfachheit und vielfältige Anwendung unter einen Hut bringen
Projektphase	Konzept, Anforderungserhebung
Methoden	Personas und Szenarien, UI Prototyping, Design Workshops, Use Cases

Ausgangslage

Ein Hersteller von Schneidemaschinen ist gut im High-end-Bereich positioniert und will mit einem neuen Produkt in ein neues Marktsegment vordringen. Dieses soll insbesondere auch bereits für kleine Firmen mit zwei bis drei Angestellten erschwinglich sein. Mit der Maschine kann eine solche Firma Kleinserien von Blechteilen in einer fast beliebigen Form aus einem Blech schneiden, beispielsweise für Lampen, Zahnräder, Luftgitter, Halterungen und vieles mehr. Die Entwicklung der Maschine selber ist bereits gut vorangeschritten, und es geht nun darum, die grafische Benutzerschnittstelle des Bedienpanels zu definieren. Die folgende Abbildung skizziert ein solches Bedienpanel für eine Schneidemaschine.

Herausforderungen

- Der internationale Vertrieb der Maschine erzeugt eine große Distanz zu den Benutzern, nicht nur geografisch, sondern auch unternehmensbezogen: Der Hersteller kommuniziert mit den Kunden nur indirekt über die lokalen Vertriebsgesellschaften.
- Das Bedienpanel besteht nicht nur aus einem GUI, sondern ist auch ein Stück Hardware und Teil einer Maschine. Die Funktionsweise der Maschine beeinflusst das Panel und umgekehrt.
- Für Usability Engineering waren nur wenige Tage Aufwand budgetiert.



Vorgehen

Der Einsatz des Usability Engineer fokussierte sich in diesem Projekt auf die Phase der Anforderungserhebung. Einfache Personas halfen, das Zielpublikum dem Projektteam näher zu bringen. Die Grundlagen waren Informationen, welche von den lokalen Vertriebsgesellschaften mittels Fragebogen eingeholt wurden, sowie einzelne Interviews mit Servicetechnikern und Trainern, welche gelegentlich beim Kunden vor Ort sind. Die Personas räumten mit vielen Vorurteilen auf. So entdeckte das Projektteam nicht den oft zitierten, angelernten Reisbauern, sondern einen gut ausgebildeten Ingenieur mit einem Bachelor-Abschluss. Passend dazu zeigten die Personas auch, dass die vorgesehene, strikte Rollenaufteilung zwischen dem Einrichter, welcher Aufträge auf der Maschine konfiguriert und dem Operateur, welcher die Aufträge dann ausführt, für dieses Modell nicht sinnvoll war.

Parallel zu dieser Erhebung erstellte das Team ein User Interface Storyboard. Dieses zeigte ein charakteristisches Anwendungsszenario mit der Maschine in allen Details auf.

Damit waren die Grundlagen für einen Workshop geschaffen. Das Ziel war, das User Interface des Bedienpanels zu definieren. Der Weg dazu führte jedoch über den Produktionsprozess als Ganzes: Aufträge definieren, Rohmaterialien zuliefern und einführen, Produktion starten und überwachen und gefertigte Teile und Abfälle abtransportieren – inklusive Fehlerfälle. Es herrschte ein reges Interesse von allen

an der Entwicklung beteiligten Disziplinen: Entscheidungsträger der Mechanik, Elektronik, Systems Engineering, Marketing und mehr waren eingeladen worden und kamen, bereit für eine aktive Diskussion. Diese Teilnehmer spielten nun mit den Personas das Szenario Schritt für Schritt durch. Ein Styropormodell der geplanten Maschine stellte die Bühne dar, und das User Interface Storyboard das Drehbuch. Mit guten Resultaten: Nicht nur wurde das Konzept für das Bedienpanel in vielen Teilen weiterentwickelt und den Personas und den Prozessen angepasst, auch die Maschine selber wurde verbessert.

Die Erkenntnisse des Workshops wurden vom Projektteam konsolidiert. Mit der überarbeiteten Version wurde anschließend ein zweiter Workshop gestaltet, in welchem auf bisher noch offen gebliebene Fragestellungen eingegangen wurde und die Arbeiten bestätigt wurden. Auf Basis der so entstandenen Szenarien und User Interface Storyboards war das Dokumentieren der Anforderungen für das Projektteam nun auch sehr einfach. Use Cases und Skizzen des User Interfaces wurden dazu verwendet.

Der Usability-Experte begleitete das Team des Bedienpanels in der Methodik, leitete insbesondere die beiden Workshops und gab regelmäßig Feedback zu den vom Team erstellten Mock-ups. Die so erstellte Spezifikation wurde anschließend umgesetzt.

Nutzen und Fazit

- Bei einer Maschine für die industrielle Produktion geben Hardware und mögliche Produktionsprozesse das User Interface des Bedienpanels stark vor. Die Usability-Diskussion fokussierte konsequent auf die Prozesse und die Hardware und gestaltete dafür das Bedienpanel so einfach wie möglich; mit dem Nebeneffekt, dass auch Optimierungen bei Maschine und Prozessen vorgenommen wurden.
- Die Käufer der Maschine und deren Mitarbeiterstruktur definierten insbesondere Rollenaufteilung und Ausbildungsniveau der Benutzer. Die Personas brachten diese beiden Punkte in die Diskussion ein. Sie schärften den Blick auf das Zielpublikum, und es gelang, einige Vorurteile zu lösen. Einfachere und direktere Abläufe auf dem Bedienpanel waren die Folge davon.
- User Interface Prototyping mit Papier und mit der Entwicklungsumgebung erlaubte es, in kurzer Zeit mehrere Iterationen des Bedienpanels durchzuführen. Anpassungen an Personas und Produk-

tionsprozess, wie auch das Feedback vom Gesamtprojektteam und dem Usability-Experten konnten so sehr kostengünstig eingearbeitet werden.

- Diesem Projekt fehlte eine Feedbackschleife mit tatsächlichen Benutzern. Eine vertiefte Analyse mit Contextual Inquiry wie auch Usability-Tests mit Benutzern hätten sicherlich noch viele wichtige Punkte zu Tage gefördert. Doch ein solches Vorgehen hätte das Budget für Usability Engineering weit überzogen.
- Für Hersteller und Kunden stimmte der gewählte Kompromiss: Der Hersteller erhielt mit wenig Zusatzaufwand ein Bedienpanel mit einer im Vergleich zur Konkurrenz guten Usability und die Kunden eine leistungsfähige Maschine zu einem guten Preis, mit welcher auch die Mitarbeiter gut arbeiten konnten.

Schwierige Situationen

Hoffentlich sprühen Sie nun vor Tatendrang und möchten das Gelesene baldmöglichst umsetzen. Manchmal holt einen die Praxis allerdings nur allzu schnell wieder auf den Boden zurück. Die Landung ist zuweilen hart, doch selten schädlich. In diesem Abschnitt stellen wir in der Praxis häufig anzutreffende, schwierige Situationen dar und zeigen Ansätze, diese zu bewältigen.

Benutzer werden vertreten

In größeren Unternehmen sind **Benutzervertreter** oft fester Bestandteil eines Projektteams. Dies ist grundsätzlich eine ausgezeichnete Idee, da diese Fach und Anwendung kennen und entsprechend wichtige Punkte schnell klären können.

Die Schwierigkeit in der Praxis besteht darin, dass Benutzervertreter oft mit der Aufgabe betraut werden, selbst die Anforderungen an die Benutzerschnittstelle für ein neues System zu definieren. Benutzervertreter sind allerdings weder Methodiker noch Systemanalysten noch User Interface Designer und auch keine Software-Architekten. Es ist für sie deshalb kaum möglich, eine passende Lösung zu definieren. Es besteht die Gefahr, dass zentrale Anforderungen vergessen und persönliche Wünsche eingefordert werden, die den Benutzern, die sie vertreten, nicht dienen. Mit dieser Arbeit verlieren sie mit der Zeit auch den unvoreingenommenen Blickwinkel der Benutzer und können das System nicht mehr aus deren Sicht beurteilen. Die Problematik ist

in jenen Firmen besonders stark ausgeprägt, in welchen Mitarbeiter der Geschäftseinheiten vollamtlich als Benutzervertreter eingesetzt werden und deshalb bald keine eigentlichen Benutzer mehr sind.

Die Projektleitung kann in einer solchen Situation eine entscheidende Rolle übernehmen. Es liegt an ihr, die notwendige Methodik einzuführen und ein Projektteam mit dem benötigten Know-how zusammenzustellen. Im Idealfall bildet sie ein effektives Team von Benutzervertretern. Dieses Team sollte aus erfahrenen Benutzern bestehen und eine große fachliche Bandbreite abdecken. Es nimmt die folgenden Aufgaben wahr:

- Bedürfnisse der Anwender und Geschäftseinheiten einbringen
- Fachliche Fragen beantworten
- Den Zugang zu Benutzern ermöglichen
- Das System auf fachliche Korrektheit testen

Dieses Team bildet die Schnittstelle zwischen Entwicklung, Benutzern und Fachleuten. Es ist für die korrekte Bedürfnisabklärung bei den Benutzern mitverantwortlich. Zur Erfüllung dieser Schnittstellenfunktion helfen die Benutzervertreter bei Contextual Inquiry mit, unterstützen bei der Erarbeitung von Personas, Szenarien und Storyboards, helfen beim User Interface Prototyping und rekrutieren Testpersonen für Usability-Tests. Damit die Benutzervertreter diese Aufgaben lösen können, müssen sie entsprechend methodisch unterstützt werden.

Kein Benutzer am Horizont

Bei manchen Projekten bleibt der Zugang zu Benutzern verwehrt. Damit fehlt dem Analysten eine wichtige Informationsquelle zur Erarbeitung einer benutzbaren Lösung. In dieser Situation können die folgenden sekundären Quellen hilfreich sein:

- Lokale Vertriebsgesellschaften
- Hotline und Support
- Trainer
- Schulungsmaterial und Bedienungsanleitungen
- Serviceleute
- Konkurrenzprodukte
- Verfügbare Literatur
- Vorhandenes Wissen über die Benutzer im Unternehmen, wie beispielsweise aus Marketing und Marktforschung
- Kundenkontakte auf Messen und Ausstellungen

Methoden wie Personas und Szenarien tragen dazu bei, das vorhandene Wissen der Beteiligten für die Produktvision zu nutzen sowie verborgene Annahmen über Benutzer und Anwendung auf den Tisch zu bringen, um ein gemeinsames Verständnis zu erarbeiten.

Es kann für ein Unternehmen lohnenswert sein, das Sammeln solcher Informationen zu institutionalisieren und damit einen ständigen Draht zu Benutzern und Kunden zu etablieren. Beispielsweise könnte es zu der Tätigkeit von Service-Ingenieuren gehören, vor Ort gezielt Informationen über die Benutzer aufzunehmen, die zur Weiterentwicklung der Produkte beitragen.

Wenn Zeit und Geld knapp sind

Im Kapitel „Usability im Griff: Planung“ (S. 77) haben wir argumentiert, dass in erster Linie Geschäftsziele und Produktrisiken als Grundlage für Planung und Einsatz von Usability Engineering im Projekt dienen. In der Praxis sind Geld und Zeit meist knapp. Eine den Zielen und Risiken angepasste Methodik findet im Budget oft keinen Platz. Im Kapitel „Strategische Usability“ (S. 86) haben wir bereits den Trade-off zwischen Aufwand, Zeit, Qualität und Funktionsumfang aufgezeigt. Ist hohe Qualität gefordert, beispielsweise bei hohen Produktrisiken, führt kein Weg an der entsprechenden Methodik vorbei. Alles andere wäre schlichtweg fahrlässig.

Steht hingegen tatsächlich eine möglichst kurze Zeit bis Einführung bei begrenztem Budget im Vordergrund, dann ist ein Vorgehen erwünscht, um die schlimmsten Usability-Probleme kostenneutral auszumerzen. Für ein solches Vorgehen prägte Nielsen schon früh den Begriff *Discount Usability Engineering* [Nielsen 93]. Sein Argument war, dass mit simplen, kostengünstigen Methoden in einfachen Praxissituationen bereits relativ viel erreicht werden kann, und dass ein vereinfachtes Vorgehen auch mit höherer Wahrscheinlichkeit zur Anwendung kommt als eine vollständigere, aber auch aufwändigere Methodik. Ein solches vereinfachtes Vorgehen kann folgendermaßen aussehen:

- Minimale Analyse des Nutzungskontexts: Wenige Beobachtungen oder Interviews bei den Benutzern vor Ort anstelle einer ausführlichen Contextual Inquiry
- LoFi-Prototyping entlang der wichtigsten Anwendungsszenarien
- Iteratives Prüfen und Verbessern mittels einfacher Usability Walk-throughs mit Benutzern
- Experten-Reviews anhand von Usability-Prinzipien oder Checklisten

Wasserdicht spezifizieren?

Der Vertrag zwischen Auftraggeber und Hersteller regelt die Grundlagen, falls die Entwicklung einer neuen Lösung extern vergeben wird. Vor allem im Rahmen von Ausschreibungen werden oft Verträge ausgearbeitet, die neben Angaben zu Kosten, Zeit und Qualität auch detaillierte Vorgaben zu Funktionsumfang und Benutzerschnittstelle enthalten. Ein solch enges Korsett erlaubt nach Vertragsabschluss typischerweise nur noch kleine Optimierungen bezüglich Usability, beispielsweise Anpassungen der Benutzeroberfläche von eher kosmetischer Natur. Entsprechen Funktionalität, Informationen und Abläufe nicht den Benutzerbedürfnissen, dann sind Änderungen nach Vertragsabschluss nur noch mit größerem Aufwand möglich: interne Abstimmungen und Vertragsänderungen kosten für alle Beteiligten viel Zeit, Geld und Nerven.

Um in einer solchen Vertragssituation trotzdem die erwünschte Usability für das geplante Produkt zu erreichen, sind die Phasen bis zum Vertragsabschluss entscheidend. Der Auftraggeber sollte insbesondere auf die folgenden Punkte achten:

- Usability-Methoden für die Analyse, Modellierung, Spezifikation und Evaluation mit Benutzern sind *vor* der Auftragsvergabe durchzuführen.
- Der Auftraggeber muss dazu Usability Engineering einplanen und entsprechendes Know-how aufbieten.
- Im Vertrag muss die Aufgabenteilung mit dem Hersteller bezüglich Ausgestaltung der Benutzerschnittstelle geregelt werden. Auch die Teilung von Folgekosten, die sich aus falschen Annahmen oder fehlenden Informationen bezüglich Usability-Aspekten ergeben, kann vertraglich festgehalten werden.
- Die Spezifikation sollte dabei nicht nur vollständig ausgearbeitete Use Cases, sondern auch ein mit Benutzern geprüftes User-Interface-Konzept und gegebenenfalls UI-Prototypen und Styleguides umfassen.

Ein grundsätzlich anderer Ansatz besteht darin, *nicht* vollständig zu spezifizieren und eine enge Zusammenarbeit mit dem Hersteller anzustreben. Anstelle einer umfassenden Anforderungsspezifikation steckt der Auftraggeber seine Ziele und Rahmenbedingungen ab und definiert den Prozess für das weitere Vorgehen. Hersteller und Auftragnehmer führen Usability-Methoden gemeinsam durch. Die enge Zusammenarbeit erlaubt beiden, gegenseitige Stärken auszunutzen.

Verfügt der Auftraggeber über wenig Erfahrung im Bereich Usability Engineering, kann er dies bei der Wahl eines geeigneten Auftragnehmers berücksichtigen und sich so das notwendige Know-how sichern.

„Karl ist zuständig“

Wer ist zuständig für Usability? In der Praxis sind verschiedene Situationen anzutreffen. Typischerweise ist niemand zuständig. In der Konsequenz kümmert sich der User-Interface-Entwickler ein wenig um die Usability des Systems, der Software-Architekt hat auch eine Meinung dazu, und die Personen aus den Geschäftseinheiten geben ihre Wünsche bekannt. Usability wird in diesen Fällen eher zufällig oder gar nicht berücksichtigt.

Manchmal ist indessen eine bestimmte Person, beispielsweise Karl, explizit für Usability zuständig. Leider ist Karl auf sich allein gestellt und kämpft gegen die Wünsche der Auftraggeber, die fixen Ideen der Benutzer und die technische Verliebtheit der Software-Entwickler.

Usability Engineering ist ein Ansatz, um bestimmte Ziele zu erreichen und Risiken zu reduzieren. Es genügt nicht, dass der Designer ein geeignetes User Interface definiert, wenn der Software-Architekt eine zu langsame Architektur wählt und der Software-Entwickler unverständliche Fehlermeldungen programmiert. Damit die vorgesehene Usability erreicht werden kann, müssen viele Personen zusammenspielen. Die folgende Tabelle gibt einen Einblick, welche Rollen für welche Tätigkeiten zuständig sind:

Risikomanager	Identifizieren von Produktrisiken Definieren der Usability-Maßnahmen
Projektleiter	Einplanen von Usability Engineering Fördern des Bewusstseins für Usability Beschaffen des notwendigen Usability Know-hows Rekrutieren der Benutzer für Workshops und Usability-Tests
Analyst	Analysieren und beschreiben von Benutzern und Kontext Analysieren der Projektziele und Risikomaßnahmen bezüglich Usability-Aspekten Modellieren passender Anforderungen Evaluation der Ergebnisse mit Benutzern Vermitteln bei widersprüchlichen Anforderungen
Geschäftseinheiten	Anpassen der Organisation, Prozesse, Arbeitsanweisungen und Arbeitsplätze der Benutzer

User Interface Designer	Entwerfen einer funktionalen und ästhetischen Benutzerschnittstelle Optimieren der Benutzeroberfläche Anwendung von Styleguides
Software-Architekt	Identifizieren der architekturkritischen Anforderungen Entwerfen einer passenden Architektur
Entwickler	Implementieren der Benutzerschnittstelle
Usability-Tester	Einplanen der Usability-Tests Festlegen der Testziele Ausführen und auswerten der Usability-Tests
Technical Writer	Erarbeiten des Hilfesystems sowie der Bedienungsanleitungen

Diese lange und sicher nicht vollständige Liste illustriert eindrücklich, wie viele Stellen auf die Usability eines neuen Produktes oder zu entwickelnden Systems Einfluss nehmen. Egal ob Karl für Usability zuständig ist, in Tat und Wahrheit ist jede in einem Projekt involvierte Person mitverantwortlich, inklusive Auftraggeber und Management. Gutes Usability Engineering beinhaltet deshalb auch moderierende und vermittelnde Tätigkeiten, achtet auf gute Kommunikation und findet selten im stillen Kämmerchen statt.

Rückblick

Das also war des Pudels Kern!
(Johann Wolfgang von Goethe)

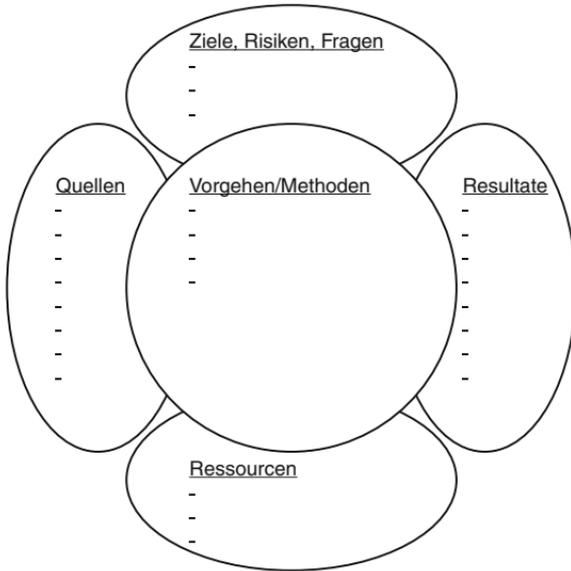
Usability Engineering folgt einer bestimmten Philosophie. Dieses Kapitel soll einige Kernpunkte nochmals in Form von fünf zentralen Prinzipien zusammenfassen. Es kann hilfreich sein, sich diese Prinzipien in einer ruhigen Minute zu vergegenwärtigen und den eingeschlagenen Kurs zu überprüfen.

Fragestellung: Zielgerichtet vorgehen

Lassen Sie die in diesem Buch vorgestellten Methoden und Konzepte Revue passieren. Es wurde oft betont, wie substanziell die Definition einer Fragestellung ist:

- Contextual Inquiry wird von den offenen Fragen gesteuert, die es zu beantworten gilt.
- Personas werden unterschiedlich charakterisiert, je nachdem, ob hohe Qualität der Arbeit, hohe Effizienz oder hohe Sicherheit im Vordergrund steht.
- Aus der Fragestellung lässt sich ableiten, welche Art von Prototyping zielführend ist: Sollen Anforderungen geklärt oder die Benutzerschnittstelle konzipiert werden? Geht es darum, ein User Interface zu optimieren oder für die Entwicklung zu spezifizieren?
- Wie und wann ein Fragebogen zum Einsatz kommt, hängt von der Fragestellung der Untersuchung ab.

Jeder Methode sollte eine wohlüberlegte Fragestellung zugrunde liegen. Die folgende Abbildung zeigt ein Hilfsmittel, um in einem Workshop eine Fragestellung zu entwickeln. Die Teilnehmer schreiben aktuelle Fragen und Ziele auf und ergänzen notwendige Quellen, Methoden, beabsichtigte Resultate und dafür erforderliche Ressourcen.



Kontext: Für die Realität entwerfen

Usability hängt per Definition nicht vom Produkt allein, sondern von der Wechselwirkung zwischen Benutzer, Werkzeug, Aufgabe und Umfeld ab. Usability-Methoden zielen deshalb darauf ab, den realen Anwendungskontext einzubeziehen:

- Im Rahmen von Contextual Inquiry geht der Analyst zu den Benutzern und beobachtet diese bei der Arbeit.
- Personas und Szenarien bringen den Kontext in Design und Entwicklung hinein.
- Storyboards zeigen das Produkt im Umfeld der Benutzung.
- In Usability-Tests werden realistische Szenarien durchgespielt.

Eine Software ist kein in sich abgeschlossenes System, sondern untrennbar mit dem Kontext verbunden. Die zu erstellende Lösung muss in der Wechselwirkung mit ihrer Umwelt betrachtet werden, damit sie auch in diese Umwelt hineinpasst.

Partnerschaft: Benutzer konstruktiv einbeziehen

Partnerschaft bedeutet, Benutzer als Experten ihres Fachgebietes zu verstehen und ihre Wünsche und Bedürfnisse zu berücksichtigen.

Benutzer kennen die konkreten Details und Besonderheiten ihrer täglichen Arbeit und sind schließlich die Messlatte für die Praxistauglichkeit des neuen Systems.

- Contextual Inquiry ermöglicht, die Arbeit der Experten vor Ort kennenzulernen.
- Storyboards, UI-Prototypen und Styleguides schaffen eine Sprache, die Benutzern und Entwicklern verständlich ist und ermöglichen die Zusammenarbeit.
- In Usability-Tests setzt das Projektteam seine Arbeit systematisch der Kritik der Benutzer aus.
- Fragebögen erlauben die Beurteilung eines Systems aus Sicht der Benutzer.

Projektmitarbeiter und Benutzer bilden ein interdisziplinäres Team und bearbeiten zusammen eine ausgewählte Fragestellung. So wird konstruktive Zusammenarbeit möglich.

Fakten interpretieren

Fakten werden zusammengetragen, interpretiert und die eigene Interpretation wiederum geprüft:

- Contextual Inquiry und Fragebögen dienen zum Sammeln und Auswerten von Fakten.
- Personas und Szenarien werden nicht einfach erfunden, sondern aus den gesammelten Informationen herausgearbeitet.
- Use Cases werden aufgrund der Ergebnisse aus der Analyse von Benutzern und Kontext erstellt.
- User-Interface-Prototypen werden in Usability-Tests objektiv geprüft.

Es sollten möglichst unvoreingenommene Diskussionen über die Software geführt werden. Projektmitarbeiter müssen vorgefasste Meinungen überprüfen und gegebenenfalls revidieren. Lassen Sie die gesammelten Fakten sprechen, hören Sie aufmerksam zu, und entwerfen Sie die zu den Fakten passende Software.

Modellieren: Entwerfen und Feedback einholen

Eine benutzbare Lösung zu entwickeln bedeutet, Entwürfe und Modelle zu erarbeiten und mithilfe der Benutzer zu überprüfen und zu optimieren:

- Contextual Inquiry nimmt Bezug auf bereits vorhandene Systeme und Arbeitsprozesse. Die Benutzer werden nicht einfach nach ihren

Vorstellungen befragt. Schon in den ersten Interviews schaffen Entwürfe und Prototypen oder vergleichbare Produkte einen Bezug zu der Erfahrungswelt der Benutzer.

- Personas und Szenarien modellieren Bedürfnisse und Anforderungen der Benutzer. In Storyboards und User-Interface-Prototypen werden diese so aufbereitet, dass die Benutzer eine reale Vorstellung von der geplanten Lösung erhalten.
- Use Cases repräsentieren die geplante Funktionalität des neuen Systems. Sie werden aufgrund der Rückmeldungen der Beteiligten präzisiert.
- User-Interface-Prototypen spiegeln die geplanten Abläufe aus Benutzersicht wider. In Usability Tests wird die Interaktion geprüft und angepasst.

Das Verständnis des Modellierens ist im Usability Engineering essenziell. Ein effektiv und effizient benutzbares Produkt kann nicht auf Anhieb erstellt werden. Feedbackzyklen auf verschiedenen Ebenen sind notwendig.

Denkanstoß: *Halten Sie sich ein aktuelles Projekt vor Augen. Wenn Sie die fünf geschilderten Prinzipien auf die methodischen Gepflogenheiten in diesem Projekt anwenden würden, was müssten Sie ändern?*

Ausblick

Im letzten Kapitel dieses Buches geben wir Ihnen einen Überblick über einige verwandte Gebiete und Themen. Eine scharfe Abgrenzung, wo Usability Engineering aufhört und ein anderes Gebiet anfängt, ist dabei mit Sicherheit unmöglich und auch nicht unsere Absicht. Dieses Kapitel soll vielmehr aufzeigen, welche fachlichen Strömungen es gibt, wohin sich die Disziplin entwickelt und unter welchen Begriffen Sie zu einem bestimmten Thema weitere Informationen finden können.

Accessibility

Das Fachgebiet Accessibility setzt sich mit der Frage auseinander, wie technische Systeme für Menschen mit Behinderungen oder Einschränkungen zugänglich gemacht werden können. Behindertengerechte Technologien sind besonders bei öffentlichen Systemen wie beispielsweise Fahrkartenautomaten, staatlichen Webseiten, Auskunftssystemen oder Internet-Anwendungen von großer Relevanz.

Ein wichtiger Grundsatz beim Thema Accessibility ist zunächst das Prinzip der Gleichstellung. Menschen, die beispielsweise von visuellen, kognitiven oder motorischen Einschränkungen betroffen sind, sollen die Möglichkeit erhalten, technische Systeme weitgehend genauso zu nutzen wie nicht Behinderte. Moderne Technologien tragen gerade für diese Personen dazu bei, ein selbstständiges Leben zu führen. So erlauben etwa elektronische Publikationen auch Blinden oder stark Sehbehinderten die Konsumation von schriftlichen Medien. Online-Anwendungen ersparen den Weg zur Bank oder ins Lebensmittelgeschäft. Gleichstellungsgesetze sind für eine ganze Reihe von Ländern in Kraft, dazu gehören auch Deutschland, Österreich und die Schweiz.

Große Bedeutung erlangte die Einhaltung von Accessibility-Kriterien in der Entwicklung von Webseiten. Im Rahmen der Standardisierung von Web-Technologien wurden Accessibility-Standards verabschiedet [W3C 07], die beispielsweise die freie Vergrößerung von Schriften oder die Ausgabe von Inhalten mit Hilfsmitteln wie Braille-Lesegeräten und Sprachausgabesystemen sicherstellen. Im deutschsprachigen

Raum hat sich dafür der Begriff *barrierefreies Internet* eingebürgert. Webseiten der öffentlichen Hand müssen barrierefrei gestaltet sein.

User Experience

Die noch relativ junge Disziplin User Experience hat zum Ziel, das Gesamterlebnis zu optimieren, das mit der Nutzung von Diensten oder Produkten zusammenhängt. Ihre Vertreter führen ins Feld, dass sich Usability zu stark auf die funktionalen Aspekte technischer Systeme bezieht, und versuchen, die Betrachtungsweise um mehrere Dimensionen zu erweitern:

- Neben der Benutzbarkeit technischer Systeme sollen auch Erlebnisse mit nicht technischen Systemen optimiert werden, wie beispielsweise Einkaufswelten, Museen, Bibliotheken, Messen und ähnliche Einrichtungen.
- Insbesondere in der Interaktion mit Kunden gilt es, das Erlebnis über alle Kundenberührungspunkte des Unternehmens (*Customer Touch Points*) zu verbessern, also beispielsweise alle Kontakte eines Kunden vor und nach dem eigentlichen Bestellprozess. Dazu gehören das Einholen von Informationen ebenso wie die Unterstützung während Bestellung und Kauf oder Supportanfragen bei Problemen.
- Der Fokus der Optimierungen liegt nicht allein auf funktionalen Aspekten, sondern es werden vermehrt auch Faktoren wie Design und Ästhetik einbezogen, welche das Vergnügen in der Benutzung erhöhen sollen. Das aktuelle Schlagwort dafür ist *Joy of Use*.

Das Thema User Experience bewegt sich damit nahe am Dienstleistungsmarketing, welches ebenso zum Ziel hat, dem Kunden eine ideale Erlebniswelt zu vermitteln.

Interaction Design

Das Interaktionsdesign definiert die Möglichkeiten zur Steuerung eines Systems, dessen Verhalten sowie Rückmeldungen an den Benutzer. Als Fachgebiet fokussiert sich Interaction Design auf die Herleitung und den Entwurf von Interaktionskonzepten für digitale Produkte und Anwendungen. Neben der Gestaltung der funktionalen Benutzerschnittstelle (User Interface Design) steht dabei eine gute User Experience im weiteren Sinne im Vordergrund, welche das Ge-

samerlebnis bei der Verwendung des Produktes umfasst. Usability Engineering, Interaction Design und User Interface Design sind somit in der Regel eng gekoppelt und nicht scharf trennbar. Mittels Usability-Methoden werden die notwendige Funktionalität, Informationen und Abläufe definiert und mit Benutzern überprüft. Im Interaction Design werden die Bedürfnisse der Benutzer in konkrete Entwürfe der Benutzerschnittstelle umgesetzt, und im User Interface Design werden die Konzepte auf der Zielplattform implementiert.

Das Thema Interaction Design wurde in den vorigen Kapiteln schon mehrmals gestreift:

- Im Abschnitt „UI Prototyping“ (S. 40) haben wir gezeigt, wie Prototypen unter anderem zur Konzeption der Benutzerschnittstelle, zur Optimierung von Bedienelementen und zur Ausarbeitung des Designs beitragen können.
- Der Abschnitt „Usability Guidelines und Styleguides“ (S. 54) gibt einen Überblick über Richtlinien und Vorgaben für die visuelle Gestaltung und das Layout einer bestimmten Benutzeroberfläche.

Interaction Design beinhaltet die folgenden wesentlichen Aspekte:

- Kenntnisse über gute Gestaltung und Designprozesse mit Fokus auf interaktive Produkte, z. B. aus den Disziplinen der grafischen Gestaltung oder dem Industriedesign
- Kreative, gestalterische Techniken und Aktivitäten unter Berücksichtigung der zu vermittelnden Gefühlswelten
- Detaillierte Kenntnisse der eingesetzten Technologie der Benutzerschnittstelle

Die Art der Benutzerschnittstelle ist dabei von zentraler Bedeutung. Während die eingesetzten Usability-Methoden für unterschiedliche Technologien sehr ähnlich oder sogar gleich ablaufen, ist es für den Entwurf der Benutzerschnittstelle ein wesentlicher Unterschied, ob beispielsweise ein Datenverarbeitungssystem, ein mobiles Gerät, ein Bedienpanel zur Maschinensteuerung oder ein Fahrkartenautomat entwickelt wird. Folgende Punkte müssen unter anderem berücksichtigt werden:

- Eingabe- und Ausgabemedien wie Maus und Tastatur, Bildschirme, Touchscreens, Tasten und Displays auf Geräten
- Zu verwendende Bedienelemente
- Struktur und Layout der Benutzerschnittstelle
- Aspekte wie Ergonomie, grafische Gestaltung, Industriedesign, technische Umsetzbarkeit, Corporate Design und Ästhetik

Zu empfehlende Werke zum Thema Interaction Design sind [Cooper & Reimann 07] sowie [Buxton 07].

Security und Usability

Das Fachgebiet der IT-Security befasst sich damit, Systeme und die damit bearbeiteten Informationen vor unerlaubtem Zugriff zu schützen. Klassischerweise befassen sich Experten dabei vor allem mit den technischen Mitteln, wie beispielsweise Verschlüsselung, Identifikationsverfahren, Virenschutz, Firewalls usw.

Immer öfter melden sich zum Thema Sicherheit auch Fachleuten aus dem Bereich Usability zu Wort. Ein System kann nur sicher sein, wenn die Benutzer auch richtig mit der Sicherheitstechnologie umgehen. Beispiele für falsche Verwendung kennen wir aus unserem eigenen Alltag: Wir verschicken vertrauliche Dokumente unverschlüsselt per E-Mail, verwenden immer wieder dieselben Passwörter oder notieren sie, öffnen potenziell böswillige E-Mails und so weiter.

Damit Sicherheitstechnologien auch verwendet werden, müssen sie einfach und effizient sein und dürfen die Arbeit nicht behindern. Eine aus technischer Sicht gesteigerte Sicherheit, die Teamwork, Mobilität oder Effizienz der Benutzer in unzumutbarer Weise einschränkt, wird ganz einfach umgangen.

Das Gebiet Security setzt sich deshalb immer mehr auch mit den menschlichen Verhaltensweisen bei der Anwendung von Technologien auseinander. Usability-Methoden helfen dabei, die tatsächlichen Ziele und Handlungsweisen der Benutzer zu verstehen. Entsprechend erscheinen vermehrt Publikationen, die sich mit der gegenseitigen Abhängigkeit von Security und Usability befassen. Eine umfassende Darstellung bietet das Buch *Security and Usability* [Cranor & Garfinkel 05].

Web Usability

Nach dem Internetboom Ende des letzten Jahrtausends setzte sich bei vielen Unternehmen die Erkenntnis durch, dass Websites nur dann eine Wirkung erzielen, wenn sie auch bezüglich Usability einen guten Stand aufweisen. Es erschienen zahlreiche Publikationen mit Vorgehensweisen, Richtlinien, Tipps und Tricks für die benutzerfreundliche Gestaltung von Web-Auftritten (siehe auch Abschnitt „Usability

Guidelines und Styleguides“, S. 54). Web Usability wurde beinahe zum Synonym für gutes Web Design. Die Benutzbarkeit von Websites wird durch einige Besonderheiten beeinflusst:

Websites stellen meist eine große Menge von Informationen dar, in denen sich ein Besucher zurechtfinden soll. Eine wichtige Frage ist deshalb, wie Benutzer durch das Web-Angebot navigieren. Ein zentraler Aspekt ist dabei der Aufbau einer verständlichen und übersichtlichen *Informationsarchitektur*.

Auch wenn Grafiken und Bildwelten schon lange im Webdesign Einzug gehalten haben, sind Texte nach wie vor das wichtigste Medium zur Vermittlung von Informationen und Inhalten. Web Usability beschäftigt sich deshalb mit der Darstellung und Aufbereitung von Texten für die Verwendung im Web.

Bei Web-Auftritten treffen zwei unterschiedliche Interessen aufeinander: Zum einen das Ziel der Besucher, Informationen und Inhalte effizient zu finden, und zum anderen die Absicht der Marketingabteilungen, die Identität ihres Unternehmens zu kommunizieren. Für Web Usability ist deshalb auch ein Verständnis für effektive Kommunikation mit Text und Bild von Bedeutung. Zum Thema Web Usability und Web Design gibt es viel Literatur. Ein empfehlenswertes Buch ist [Wirth 04].

Durch neue Technologien und aufgrund der zunehmenden Verbreitung von Breitbandanschlüssen erfährt das Web zurzeit die zweite große Veränderung seit dem Wandel vom universitären Wissensnetz zur kommerziellen Informationsplattform. Multimediale Inhalte, Kollaboration und Community-Plattformen sind aktuelle Trends, die unseren Alltag und unser Sozialverhalten verändern. Es warten neue Aufgaben auf die Disziplin Web Usability.

Der allgegenwärtige Computer

Das Gebiet *Ubiquitous Computing* stellt die These auf, dass die Allgegenwärtigkeit der Computertechnologie die nächste Ära der Informationsgesellschaft einläutet. Der Begriff geht auf eine Veröffentlichung des Informatikers Mark Weiser zurück [Weiser 91].

Durch die fortschreitende Miniaturisierung und Vernetzung nehmen intelligente Gegenstände mehr und mehr Einzug in unseren Alltag und lösen das Paradigma des klassischen PCs mit seinen Limitationen ab. Schon heute gibt es dafür zahlreiche Anwendungen,

wie rechnergestützte Haussteuerungen, die Verschmelzung von Computer, Videorekorder, Hi-Fi-Anlage und Fernsehgerät, den medienwirksamen intelligenten Kühlschrank mit Internetanschluss, RFID-Mikrochips zur Kennzeichnung von Gegenständen und viele mehr. In Zukunft sind noch kleinere Computereinheiten in großer Anzahl denkbar, die unsere Umgebung durchdringen.

Weiser selbst legte großen Wert darauf, dass neben den technologischen Möglichkeiten auch die damit verbundenen soziologischen Veränderungen dieser neuen Ära diskutiert werden. Ein wesentlicher Punkt dabei ist, dass die Computerisierung der Umwelt für den Menschen weitgehend unsichtbar bleibt und im Gegensatz zum klassischen Computer keine spezielle Aufmerksamkeit erfordert. Information wird prinzipiell überall dort zur Verfügung stehen, wo sie benötigt wird. Deren Abruf soll auf natürliche Art und Weise erfolgen, ohne zu einem Informationsüberschuss zu führen. Anwendungen sollen damit schneller, einfacher und intuitiver werden. Information wird auch jenen zur Verfügung stehen, die heute keinen Zugang zu entsprechenden Technologien haben.

Noch ist diese Vision mit einigen offenen Fragen verbunden. Aspekte wie Sicherheit, Privatsphäre und Datenschutz bedeuten dabei eine besondere Herausforderung.

Auch für das Gebiet Usability Engineering führt dieser Paradigmenwechsel zu Veränderungen. Individuelle, nicht standardisierte Geräte-Hardware, neue Interaktionsweisen, intelligente, mitdenkende Systeme und in die Umgebung integrierte Geräte fordern die Disziplin.

Eines wird indessen gleich bleiben: Der „Faktor Mensch“ wird bei der fortschreitenden Integration der Informationstechnologie in unseren beruflichen und privaten Alltag eine zentrale Rolle spielen. Die Analyse und das Verständnis menschlicher Ziele und Verhaltensweisen werden für die Umsetzung in nützliche und benutzbare Anwendungen immer substanzieller.

Persönliches Logbuch des Captains, Nachtrag: Wie stellt man dieses verflixte Logbuch eigentlich wieder ab?

Glossar

Analyst

Rolle im Software-Entwicklungsprozess. In diesem Buch in erster Linie Personen, die mit der Aufnahme und Modellierung von Anforderungen beschäftigt sind.

Anforderung

Eine geforderte Eigenschaft, die vom neuen System erfüllt werden soll.

Anforderungsmanagement (Requirements Engineering)

Eine Sammlung von Methoden, um die Anforderungen für die Entwicklung eines technischen Systems zu erheben, zu modellieren, zu verwalten, zu prüfen und zu kommunizieren.

Benutzer

Rolle im Software-Entwicklungsprozess. Diese sollte ausschließlich von Personen wahrgenommen werden, die potenziell das neue technische System benutzen werden.

Benutzerschnittstelle (User Interface)

Auch Mensch-Maschine-Schnittstelle, Benutzungsschnittstelle oder Benutzeroberfläche genannt. Die Benutzerschnittstelle umfasst die Teile eines Systems, mit denen Personen direkt interagieren.

Computer

Eine Maschine, mit der man fast so schnell schreiben wie denken kann. (Umberto Eco)

Ergonomie

Wissenschaft zur Erforschung der Beziehungen zwischen dem Menschen und seiner Arbeit, Arbeitsmittel und Umgebung.

Evaluation

Bezeichnung für den Vorgang, ein technisches System mit Benutzern zu beurteilen.

Geschäftsprozessmodellierung (Business Modeling)

Eine Sammlung von Methoden, um ein Modell eines Geschäftssystems zu erstellen, zu verwalten und zu überprüfen.

Human Computer Interaction (HCI)

Wissenschaft zur Erforschung der Kommunikation zwischen Mensch und Computer.

Human Factors

Psychologisches Fachgebiet zur Erforschung menschlicher Einflussgrößen bei der Anwendung von Technologien.

Interaction Design

Die Gestaltung der Interaktion zwischen Benutzer und System. Das Interaktionsdesign definiert die Möglichkeiten zur Steuerung eines Systems, sein Verhalten und seine Rückmeldungen. Als Fachgebiet umfasst Interaction Design die Herleitung einer guten Benutzerschnittstelle durch die Einbeziehung von Benutzern und beinhaltet dabei Aspekte der grafischen Gestaltung oder des Industriedesigns.

Iteratives Vorgehen

Wiederholende und zielgerichtete Durchführung eines Entwicklungszyklus von Analyse bis Evaluation zur schrittweisen Annäherung von Prototypen oder Teilen eines neuen Systems an die gewünschte Lösung.

Kontext

Auch Anwendungs- oder Nutzungskontext. Die Umwelt, in die ein technisches System eingebettet wird, bestehend aus den Benutzern und deren Arbeitsweisen sowie dem kulturellen, sozialen, organisatorischen, physischen und technischen Umfeld.

Mock-up

Eine Attrappe der Benutzerschnittstelle, die zur Evaluation mit Benutzern und anderen Interessensvertretern verwendet werden kann. Beispielsweise mit Papier, Karton, Styropor oder mit elektronischen Mitteln wie Grafikprogrammen erstellt. In diesem Zusammenhang spricht man auch von LoFi-Prototypen (von englisch *Low Fidelity*: geringe Wiedergabetreue).

Modellieren

Die vereinfachte Darstellung und Verdeutlichung eines gewissen Ausschnitts der Realität zu einem bestimmten Zweck. Modellieren ist eine notwendige Tätigkeit auf dem Weg von den Benutzerbedürfnissen zur Spezifikation eines Systems.

Prozess

Der Ablauf von Tätigkeiten, um ein bestimmtes Ziel zu erreichen. Häufig anzutreffende Bedeutungen sind: Informationsflüsse und Zusammenarbeit im Unternehmen (Geschäftsprozesse), Arbeitsabläufe eines Benutzers mit einem bestimmten System (Mensch-System-Interaktion), Vorgehensmodelle für die Erstellung von Anwendungen und Produkten (Entwicklungsprozesse).

Software-Ergonomie

Fachgebiet zur Erforschung und Anpassung von Computerprogrammen an die menschlichen Anforderungen und Bedürfnisse.

Spezifikation

Beschreibung einer neuen Lösung als Vorgabe für die Entwicklung. Teil des Vertrags zwischen Auftraggeber und Hersteller.

Usability

Ein Maß für die Benutzbarkeit eines technischen Systems. Verwandte Begriffe sind Gebrauchstauglichkeit und Benutzerfreundlichkeit.

Usability Engineering

Die systematische Anwendung von Methoden und Techniken, um gezielt benutzbare Systeme und Produkte zu entwerfen und herzustellen.

Usability Lab

Eine speziell für Usability Testing eingerichtete Räumlichkeit.

Usability-Prinzipien

Allgemeine Richtlinien für die benutzerorientierte Gestaltung von Benutzerschnittstellen. Usability-Prinzipien können sowohl für den Entwurf als auch zur Evaluation von User Interfaces eingesetzt werden.

Use-Case-Modell

Ein Modell des Verhaltens eines Systems. Die am häufigsten verwendeten Modellelemente sind Anwendungsfall (Use Case), Akteur (Actor) und Beziehung (Relationship).

User Centered Design (UCD)

Vorgehensmodelle, welche die späteren Benutzer systematisch in die Entwicklung von Systemen und Produkten einbeziehen.

User Experience (UX)

Disziplin, die sich mit der Gestaltung der gesamten mit einem Produkt oder einer Dienstleistung verbundenen Kommunikation zum Benutzer auseinandersetzt.

User Interface Design

Fachbereich, der sich mit der Gestaltung von Benutzerschnittstellen anhand definierter Anforderungen beschäftigt.

Literatur

Apple Human Interface Guidelines, 2000-2009
<http://developer.apple.com/mac/library/documentation/UserExperience/Conceptual/AppleHIGuidelines/>

Beyer, H. & Holtzblatt, K.
Contextual Design: Defining Customer-Centered Systems
Morgan Kaufmann, San Francisco, 1998

Bortz, J. & Döring, N.
Forschungsmethoden und Evaluation: für Human- und Sozialwissenschaftler
Springer, Berlin, 2006

Buxton, B.
Sketching User Experiences: Getting the Design Right and the Right Design
Morgan Kaufmann, 2007

Cockburn, A.
Use Cases effektiv erstellen
Mitp-Verlag, Bonn, 2008

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

Cooper, A. & Reimann, R.
About Face 3.0: The Essentials of Interaction Design
Wiley, Indianapolis, 2007

Courage, C. & Baxter, K.
Understanding Your Users: A Practical Guide to User Requirements Methods, Tools and Techniques
Morgan Kaufmann, San Francisco, 2005

Cranor, L. F. & Garfinkel, S.
Security and Usability: Designing Secure Systems That People Can Use
O'Reilly Media, 2005

DIN EN ISO 9241
Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten
seit 2006: Ergonomie der Mensch-System-Interaktion
Beuth, Berlin, 1997–2008

DIN EN ISO 13407
Benutzer-orientierte Gestaltung interaktiver Systeme
Beuth, Berlin, 2000

Dörnemann, H. & Meyer, R.
Anforderungsmanagement kompakt
Spektrum Akademischer Verlag, Heidelberg, 2003

EG 90/270/EWG
Verordnung über Sicherheit und Gesundheitsschutz bei der Arbeit an Bildschirmgeräten
EG, 1990

Google User Experience Principles, 2009
<http://www.google.com/corporate/ux.html>

IEC 60601-1-6
Medical electrical equipment - Part 1-6: General requirements for basic safety and essential performance - Collateral standard: Usability
IEC, 2006

IEC 62366
Application of usability engineering to medical devices
IEC, 2007

ISO/DIS 9241-210
Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems
Beuth, Berlin, 2008

Johnson, J.

GUI Bloopers 2.0: Common User Interface Design Don'ts and Dos
Morgan Kaufmann, San Francisco, 2007

Mayhew, D.

The Usability Engineering Lifecycle: A Practitioner's Handbook for
User Interface Design
Morgan Kaufmann Academic Press, San Diego, 1999

Microsoft Windows User Experience Interaction Guidelines, 2005-
2009

<http://msdn.microsoft.com/>

Miller, G. A.

The Magical Number Seven, Plus or Minus Two: Some Limits on our
Capacity for Processing Information
Psychological Review 63 (S. 81–97), 1956

Nielsen, J.

Usability Engineering
Morgan Kaufmann, San Francisco, 1993

Nokia Usability Documentation, 2010

[http://www.forum.nokia.com/main/resources/documentation/
usability/](http://www.forum.nokia.com/main/resources/documentation/usability/)

Norman, D.

The Psychology of Everyday Things
Basic Books, New York, 1988

Prümper, J. & Anft, M.

ISONORM 9241/10, Beurteilungsbogen auf Grundlage der Internatio-
nalen Ergonomie-Norm ISO 9241-10
Büro für Arbeits- und Organisationspsychologie, Berlin, 1997

Richter, M.

Online Befragung als neues Instrument zur Beurteilung der Benutzer-
freundlichkeit von Software. In U.-D. Reips, B. Batinic, W. Bandilla, M.
Bosnjak, L. Gräf, K. Moser, & A. Werner (Hrsg.), Aktuelle Online For-
schung – Trends, Techniken, Ergebnisse.

Online Press, Zürich, 1999

<http://www.gor.de/gor99/tband99/>

Rosson, M. B. & Carroll, J. M.

Usability Engineering: Scenario-Based Development of Human Computer Interaction

Morgan Kaufmann, San Francisco, 2002

Sharp, H., Rogers, Y. & Preece, J.

Interaction Design: Beyond Human-Computer Interaction

Wiley, 2007

Snyder, C.

Paper Prototyping: The fast and easy way to define and refine User Interfaces

Morgan Kaufmann, San Francisco, 2003

The Standish Group

“Chaos Report”, 1994-2009

<http://www.standishgroup.com/>

Tidwell, J.

Designing Interfaces: Patterns for Effective Interaction Design

O’Reilly Media, 2005

Vukelja, L., Müller, L., & Opwis, K.

Are Engineers Condemned to Design? A Survey on Software Engineering and UI Design in Switzerland. In: M. C. C. Baranauskas, P. A. Palanque, J. Abascal, S. D. J. Barbosa (eds.), INTERACT (2), vol. 4663 of Lecture Notes in Computer Science (S. 555–568)

Springer, 2007

W3C Web Content Accessibility Guidelines 2.0, 2008

<http://www.w3.org/WAI/>

Weiser, M.

The Computer for the 21st Century

Scientific American, 265(3), 1991

Literatur

Willumeit, H., Gediga, G. & Hamborg, K.-C.
IsoMetrics: Ein Verfahren zur formativen Evaluation von Software
nach ISO 9241/10
Ergonomie & Informatik, 27 (S. 5–12), 1996

Wirth, T.
Missing Links: Über gutes Webdesign
Hanser, München, 2004

Index

A

Accessibility 121f
Akteure 16, 49f, 52
Analyse 15f, 19, 25, 71f
Anforderung 17, 32, 41f, 51,
98, 128
Anforderungsanalyse 10, 49,
71
Anwendungsfälle 17, 48, 52f,
99
Anwendungsgebiete 6f
Aufgabenbereiche 14

B

Barrierefreies Internet 122
Benutzbarkeit 3, 130
Benutzer 15, 22, 29f, 84, 118f,
128
Benutzerbefragungen 68f
Benutzerfreundlichkeit 3, 130
Benutzergruppen 30
Benutzerperspektive 33f
Benutzerschnittstelle 42–44,
123, 128
Benutzervertreter 111
Beobachtung 22f, 64

C

Contextual Design 26
Contextual Inquiry 15, 22, 50

D

Definition 4
Discount Usability Engineering
113
Domänenmodell 17, 26

E

Entwicklungsprozess 10–20,
88f
Ergonomie 8, 128
Essential Use Cases 51f
Evaluation 18–20, 32, 61, 65,
72, 128

F

Fallstudien 96–111
Feldtests 66
Fragebögen 67–76
Fragebogenkonstruktion 70f
Funktionale Anforderungen 51
Funktionalität 6, 16, 40, 42, 48,
51, 81, 98, 100, 114

G

Gebrauchstauglichkeit 3f, 130
Geschäftsprozessmodell-
lierung 15, 27, 129
Geschäftsstrategie 92f
Gesetzliche Verordnungen 55
Glossar 17
GUI-Elemente 56, 59, 91
Gütekriterien 64f

H

Human Computer Interaction 8,
129
Human Factors 8, 129

I

Informationsarchitektur 125
Innovation 27, 104–108
Institutionalisierung 92f

Interaction Design 122–124,
129
Interview 23, 25, 63, 85
ISO-Norm 4, 12, 55, 72
Iteratives Vorgehen 18, 45f, 88,
129

K

Kontext 15f, 25, 52, 71f, 118,
129

L

LoFi Prototyping 40

M

Mensch-Computer-Interaktion 8
Messen der Usability 74
Mock-up 41, 45, 129
Modellieren 16f, 19, 119f, 130

N

Nicht-funktionale
Anforderungen 51
Normen 55

O

Objektivität 65
Online-Befragung 72

P

Paper Prototyping 44f
Personas 16, 26, 28f, 52, 97,
106, 109
Perspektivenübernahme 2, 34
Planung 77–85
Planungsbeispiele 80–84
Prototyping Tools 46, 90
Prozess 87–89, 130

Q

qualitative Studien 68, 71
quantitative Studien 68

R

Realisierung 18, 20
Reliabilität 65
Requirements Engineering 10,
13f, 16, 88, 96–100, 128
Risikobetrachtung 78
Rollen 115
RUP 10, 48

S

Security 124
Software-Ergonomie 8, 59, 130
Spezifikation 18f, 32, 44, 49f,
91, 114, 130
Standardaufgaben 61
Standardfragebögen 72f
Standardisierung 89–91
Storyboard 17, 36–39, 50, 98
Styleguide 18, 54, 56, 59, 89f
Szenarien 16, 31–33, 52, 97,
106

T

Testbericht 64
Testpersonen 62, 66f

U

Ubiquitous Computing 125
Untersuchungsdesign 69
Urteilsfehler 73
Usability 3f, 118, 130
Usability Engineering 5f, 86f,
117, 130
Usability Guidelines 18, 54, 57f

- Usability Lab 60, 63, 130
 - mobiles 66
- Usability Walkthrough 19, 65, 99, 102
- Usability-Fragebögen 19
- Usability-Methoden 14–20
- Usability-Prinzipien 55f, 130
- Usability-Test 19, 61–65, 106
- Use-Case-Modell 16, 48f, 131
- Use-Case-Spezifikation
 - 17, 50f
- User Centered Design 11, 131
- User Experience 122, 131
- User Interface Design
 - 122, 131
- User Interface Patterns 56

- User Interface Prototyping 17, 40, 45, 90f, 106, 110
- User-Interface-Konzept 17, 32, 42
- User-Interface-Prototyp 40f, 50

- V**
- Validität 65

- W**
- Web Usability 124f
- Wireframe 41

- Z**
- Ziele 77f, 117