Mary Beth Rosson and John M. Carroll

Usability Engineering: Scenario-based Development of Human-Computer Interaction

Wie der Untertitel des Buches andeutet, ist Usability-Engineering ein durch verschiedene Arten von Szenarien unterstützter Entwicklungsprozess, wobei die Szenarien als Rohmaterial für die Herstellung verschiedener Zwischenergebnisse dienen: Nutzungsanforderungen, Prototypen, Validierung, Konsensfindung, Lösungskompromisse, Unvereinbarkeiten klären usw. Alleinstellungsmerkmal dieses Buches ist die durchgängige Anwendung von Szenarien im Usability-Engineering-Prozess.

Das Buch ist aus der Praxis des Projektstudiums am Virginia Polytechnic Institute (USA) entstanden und wendet sich deshalb hauptsächlich an Lehrende und Studierende. Aber auch Industriepraktiker werden durch das Buch angeregt, ihre eigene Praxis zu reflektieren, um vielleicht neue Wege im Usability-Engineering zu gehen, denn dieser Prozess wird durchgängig von einer weiteren Methode begleitet, der Claims Analysis. Hiermit ist die Aufarbeitung der Vor- und Nachteile eines Lösungsvorschlags oder einer Vorgehensweise gemeint. Weil jedes Projekt anders ist, gibt es im Usability-Engineering nicht den einen besten Weg. Claims-Analysis anzuwenden lohnt sich, wenn (weil) es keine Design-Rezepte gibt. Die Autoren veranschaulichen an den von Ihnen benutzten Methoden, dass diese weder zu richtigen noch falschen Lösungsvorschlägen beitragen, sondern nur zu mehr oder weniger angemessenen. Szenarien liefern den Stoff, der nötig ist, um zu Kompromissen zwischen oft unvereinbaren Lösungen zu gelangen. Szenarien zwingen die am Prozess Beteiligten, genauer zuzuhören, um heraus zu finden, was für die Nutzung eines Produkts wirklich erforderlich ist. In den Szenarien kommen die Individuen der realen Welt vor. Szenarien sind in der Sprache dieser Akteure geschrieben. Die Beurteilungs-

grundlagen für das, was im Usability-Engineering-Prozess entsteht, besitzen somit hohe Validität.

Während Szenarien und Claims-Analysis sehr anschaulich dargestellt werden, bleiben die Entwicklung von Nutzungsanforderungen sowie das Prototyping eher abstrakt. Konkret ist die Analyse der Szenarien und Prototypen zum Zwecke des Verstehens von Benutzern, ihrer Probleme und Belange im Tagesgeschäft. Abstrakt ist dagegen die Darstellung des Ergebnisses dieses Verstehensprozesses. Letztlich müssen jedoch Anforderungen abgeleitet werden. Es wird gesagt, wo diese herkommen, aber nicht, welche letztlich herauskommen, damit auf ihrer Grundlage ein Produkt entsteht. Somit ist für den Praktiker nicht leicht nachvollziehbar, ob der ganze Aufwand notwendig ist, der heutzutage durch „contextual inquiry“ (Holtzblatt & Beyer) oder „scenario-based analysis“ (Rosson & Carroll) nahe gelegt wird. Die Autoren erkennen an, dass „Field work produces a wide variety of data that can be overwhelming“ (p.54). Aber es fehlt die Methodik, hieraus den Extrakt zu gewinnen. Industriepraktiker werden vermutlich genau das vermissen, was sie auch bisher bei den so genannten Aufgabenanalysen vermissen, nämlich den Übergang von der Analyse zu einem Ergebnis, mit dem Interaktions- und Informationsdesigner weiter arbeiten können. Usability-Engineering beginnt nicht voraussetzungslos, sondern setzt meist auf fachlichen Analyseergebnissen (z. B. Geschäftsprozessmodell) auf, fokussiert die gewünschte Nutzungsqualität des entstehenden Produkts und wirkt somit auf die Fachmodelle und den Systementwurf zurück. Diese Zusammenhänge werden in dem Buch nicht deutlich.

Ein Prozessmodell des Usability-Engineering wird von den Autoren nicht ge-

sondert vorgestellt. Allerdings zeigt die Reihenfolge der Kapitel „wo es lang geht“: Erst Nutzungsanforderungen entwickeln, dann Tätigkeitsentwurf als Voraussetzung für den Entwurf von Funktionen, dann Interaktions- und Informationsdesign (Oberflächenentwurf). – Praktiker werden feststellen, dass diese Reihenfolge in vielen Projekten nicht eingehalten wird. Meist fangen die Projekte mit einer systemorientierten Anforderungsanalyse an, dann Funktionsentwurf, dann Informationsdesign und die Tätigkeiten des Benutzers ergeben sich dann irgendwie aus dem abschließenden Interaktionsdesign. Wer mit dieser Praxis brechen will, wird in den Kapiteln 3 bis 5 eine Fundgrube von Argumenten entdecken, um Projektleiter von einer besseren Vorgehensweise zu überzeugen.

Die Autoren betonen, dass Usability-Engineering nicht einfach zu lernen ist, weil verschiedene Disziplinen am Prozess beteiligt sind. Es wird Stellung bezogen gegen die Designer, die Kochrezepte mögen, etwa in Form von Guidelines. Nur erfahrene Usability-Engineers können Nutzen aus Guidelines ziehen, weil sie gelernt haben, diese im Zusammenhang mit den Umständen im Nutzungskontext zu interpretieren und anzuwenden. Den Rohstoff für diese Interpretation liefern Szenarien. Der Inhalt eines Szenarios wird im Blick behalten, wenn das Für und Wider einer Empfehlung des Guidelines betrachtet wird, denn Guidelines gelten niemals isoliert oder absolut.

Das Buch regt zur Nachahmung für ein Projektstudium an Hochschulen an. Lehrende und Studierende können für ein Semester die gemeinsam zu leistende Arbeit gut gliedern, etwa dem roten Faden des Buches folgend. Vier Abschnitte hat das Projekt: Anforderungsanalyse, Entwurf, Prototyping und Evaluierung. Zur Veranschaulichung des zu lernenden Stoffes wird ein durchgängi-

ges Beispiel gewählt, ein Projekt, das tatsächlich durchgeführt worden ist.

Im Buch wird deutlich, dass Konsensfindung das Ziel der Planung eines Entwicklungsprozesses ist, in den viele Beteiligte mit unterschiedlichen Voraussetzungen hineingehen, um mit einer gemeinsam akzeptierten Lösung heraus zu kommen. Software-Entwicklung wird als Prozess der Organisation der Kompromissfindung beschrieben. Erwartungsgemäß kommt ein Produkt dabei heraus, das anders ist, als man es sich ursprünglich vorgestellt hat. Prototyping ist ein Teilprozess, der als Prozess der Lösungsfindung beschrieben wird, in dem sich die Vorstellungen über Nutzungsanforderungen klären und konkretisieren lassen. Hier werden Erfahrungen reflektiert, die heute auch im Extreme Programming gemacht werden, und dazu geführt haben, dass man mit Voraussagen über das Projektergebnis vorsichtiger geworden ist.

In einem Punkt ist das Buch ganz un-amerikanisch. Es werden europäische Entwicklungen erwähnt, die im Usability-Engineering ähnliche Wege gehen oder bereits gegangen sind. Aber besonders europäisch wirkt das Buch in seiner kritischen Stellungnahme gegenüber der Optimierung der „performance efficiency“ von Benutzern. So verzichtet das Buch bei der Bewertung der Usability ganz auf das Zählen von „key strokes“. Es wird klar gesagt, dass die von Benutzern aufgewendeten „key strokes“ beim Ausführen einer Aufgabe nicht als valider Indikator für Usability benutzt werden können. Un-amerikanisch ist das Buch auch wegen des Glossars im Anhang des Buches – ein unverzichtbarer Teil, wenn sich Vertreter mehrerer Disziplinen im Usability-Prozess verständigen wollen.

In einem anderen Punkt ist das Buch aber ganz und gar amerikanisch. Es ist bekannt, dass Amerikaner keine inter-

nationalen Normen respektieren. So ist auch dieses Buch normungsfrei. Leser, die in Normen eine Einschränkung ihres Gestaltungsspielraums vermuten (übrigens zu Unrecht, aber das wird nur im Duell mit mir geklärt werden können) werden nach der Lektüre dieses Buches und der konkreten Umsetzung in die Praxis von Projekten feststellen, dass man spielend über die in den Normen festgelegte Mindestqualität hinausgehen kann. Denn aus einem reifen Usability-Engineering-Prozess resultiert ein gebrauchstaugliches Produkt.

Dr. Wolfgang Dzida, UEP Usability Engineering Projects, E-Mail: Wolfgang@dzida.de, Internet: www.dzida.de

„Usability Engineering: Scenario-based Development of Human-Computer Interaction“. Mary Beth Rosson and John M. Carroll. Morgan Kaufmann Publishers, San Francisco, 2002, 422 Seiten. ISBN 1-55860-712-9