

Welche Unterstützung wünschen Softwareentwickler beim Entwurf von Bedienoberflächen?

Richard Oed, Anja Becker, Elke Wetzenstein
DaimlerChrysler AG, Forschung Softwaretechnologie - FT3/SP, Ulm /
Humboldt-Universität zu Berlin, Institut für Psychologie

Zusammenfassung

Die sich wandelnden Bedingungen bei der Gestaltung von Benutzungsschnittstellen durch die Zunahme softwarebasierter Lösungen führen zu veränderten Anforderungen an die Systementwickler. Dieser Entwicklungstrend erfordert eine verstärkte Unterstützung zum Aufbau und Erhalt ergonomischer Kompetenz und zur Verbesserung der Qualität von benutzerfreundlichen Software-Oberflächen. Im Rahmen des Usability-Engineering wird deshalb gefordert, ergonomische Aspekte durchgängig im gesamten Systementwicklungsprozess zu beachten. Die derzeit verfügbaren Werkzeuge für die Softwareentwicklung beschränken sich jedoch häufig auf die Unterstützung der eigentlichen Programmierfähigkeit. Um ein Unterstützungs- und Informationssystem zu entwickeln, das darüber hinaus weitere Aufgabenbereiche des Entwicklungsprozesses abdeckt, wurden Entwickler nach ihren Wünschen für ein solches System befragt. Die Ergebnisse zeigen, dass der Unterstützungsbedarf bei den Entwicklern insgesamt sehr hoch ist. Dabei wird neben Tools, die direkt der Software-Entwicklung dienen, vor allem der Zugriff auf Informationen und Werkzeuge gewünscht, die die Projektkoordination und -dokumentation betreffen. Außerdem wird konkretes ergonomisches Erfahrungswissen gegenüber allgemeinen Vorgehensmodellen bevorzugt, was bei der Gestaltung des geplanten Systems besondere Berücksichtigung finden sollte.

1 Ausgangssituation

Ohne klare Vorgaben, allein gelassen, nicht ausreichend ausgebildet, ohne richtige Unterstützung durch die Organisation und deshalb oft überfordert fühlen sich viele Entwickler moderner Benutzungsschnittstellen. Woher kommt diese Situation?

In vielen industriellen Anwendungsbereichen geht der Trend weg von Hardware-Bedienfeldern hin zu softwarebasierten Benutzungsschnittstellen. Beispiele sind etwa Systeme aus den Bereichen Unterhaltungselektronik, Haushalt („Die Mikrowelle mit Display“), Business (Kopierer, Telefone, Handys) oder Fahrzeuge (Telematiksysteme). Hier werden vor allem aus Kosten- und Wartungsgründen vermehrt Hardware- durch Software-Bedienstellen ersetzt. Zudem werden in den Software-Oberflächen vorher getrennt vorhandene Hardwarebedienfelder zusammengeführt. Das führt in der Regel zu erhöhter Funktionalität und Komplexität der Bedienung am Bildschirm. Die Notwendigkeit einer benutzerfreundlichen Gestaltung unter Berücksichtigung von Grundsätzen der Dialog-Gestaltung wächst damit deutlich.

Der schnelle Technikwandel, die rasche und vermehrte Entwicklung neuer, innovativer Bedientechnologien, Anforderungen des Gesetzgebers¹, neue Standards und Normen, Anforderungen der Benutzer und Auftraggeber verschärfen die Situation. Sie führen in vielen Fällen zu einer Überforderung der Entwickler und ganzer Entwicklungsorganisationen. Diese sind auf die Entwicklung ergonomischer Systeme noch unzureichend vorbereitet (vgl. z.B. Schoeffel 1997). Es gibt teilweise deutliche Know-how-Defizite bei der MMI-Entwicklung generell und bei der

1 Die Bildschirmarbeitsverordnung vom 20.12.1996 und die EU-Bildschirmrichtlinie über die Mindestvorschriften bezüglich der Sicherheit und des Gesundheitsschutzes bei der Arbeit an Bildschirmgeräten vom 29.5.1990.

Kenntnis von Grundlagen der Gestaltung im Detail. Dies trifft insbesondere für Bereiche zu, die bisher eher Hardwarekomponenten entwickelt haben, bei denen aber Software-Oberflächen mehr und mehr konventionelle Hardware-Panels ersetzen. Handlungsbedarf besteht unter anderem

- beim Aufbau, dem Erhalt und dem Transfer ergonomischer Kompetenz,
- bei der effizienten Zusammenarbeit von Auftraggeber, Auftragnehmer, Entwickler und Nutzer sowie
- bei der effizienten Zusammenarbeit mehrerer Organisationseinheiten / Unternehmen in Form eines Entwicklungsverbundes (Konsortium, Joint Venture).

Für konkrete innerbetriebliche Schulungsmaßnahmen im Bereich Usability Engineering wird oftmals nur wenig Geld investiert. Auch die individuelle Weiterbildung der Entwickler durch Eigeninitiative (Literaturstudium, etc.) wird durch den starken Kosten- und Zeitdruck heutiger Projekte sehr eingeschränkt.

2 Methoden zur Entwicklung von Benutzungsschnittstellen

Die Grundprinzipien und Methoden guter Software- und Systemgestaltung sind in der wissenschaftlichen Welt bekannt und in zahlreichen Büchern sehr gut aufbereitet (Card, Moran & Newell 1983, Smith & Mosier 1986, Browne 1988, Mayhew 1992, Preece 1994, Shneiderman 1998 etc.). Allerdings reicht die bloße theoretische Kenntnis in der industriellen Praxis nicht aus. Was fehlt, ist der direkte Bezug zu den Randbedingungen im konkreten Arbeits-, Projekt- und Systemkontext aus Sicht des Entwicklers. Es gibt zahlreiche wissenschaftliche Veröffentlichungen zum Thema „Methoden und Vorgehensweisen zur Entwicklung von Benutzungsoberflächen“. Die Gesamthematik wird vornehmlich unter den Stichworten „Usability Engineering“ (Nielsen 1997) und „User/Usage Centered Design“ (Constantine 1999) zusammengefasst. Diese Methoden zielen auf eine ganzheitliche und durchgängige Betrachtung ergonomischer Aspekte während der Systementwicklung ab, so dass während der Gesamtlaufzeit eines Projekts das Thema „Usability“² ausreichend berücksichtigt wird. Es sollen sowohl Auftraggeber und Auftragnehmer als auch die späteren Benutzer der Systeme frühzeitig und kontinuierlich am Gesamtprozess beteiligt werden. Ziele sind die Verbesserung der User-Interface-Entwicklung im Rahmen von Systementwicklungsprojekten sowie letztendlich die deutliche Steigerung der Qualität von Oberflächen bezüglich Benutzerfreundlichkeit und Aufgabenangemessenheit.

Generell wird *Usability Engineering* als eine Grundmenge an Aktivitäten angesehen, die idealerweise im Laufe eines Produktlebenszyklus angewandt werden (Nielsen 1997). Der Schwerpunkt von Usability-Aktivitäten liegt dabei in frühen Phasen vor dem eigentlichen Systementwurf. Wesentliche Aktivitäten sind Benutzer-, Funktions- und Aufgabenanalyse, Analyse von existierenden Altsystemen, Definition von Usability-Zielen, Generieren paralleler Gestaltungsvarianten, Benutzerbeteiligung, Anwendung vorhandener Gestaltungsregeln, Prototyping, empirische Usability-Tests und Evaluationen, iteratives Design des Endprodukts sowie Abfrage von Benutzerfeedbacks in Feldversuchen.

Usability Engineering wird sinnvollerweise im Kontext eines typischen System-Entwicklungsprozesses angewendet. Oben genannte Aktivitäten des UE werden den verschiedenen Systementwicklungsphasen als so genannte „User Interface Design Tasks“ zugeordnet (Mayhew 1992). Auf einer etwas abstrakteren Ebene wird hier zwischen den Entwicklungsphasen „Projektvorbereitung“, „Anforderungsanalyse“, „Entwurf“, „Entwicklung“ und „Installation/Inbetriebnahme“ unterschieden. Dieses grundlegende Vorgehensmodell wurde 1996 von der DaimlerChrysler-Forschung aufgegriffen, um es im eigenen industriellen Umfeld praktisch anzuwenden und zu erweitern. Daraus entstand 1999 ein erweitertes Prozessmodell für Usability Engi-

² Brauchbarkeit, Einsatzbarkeit, Verwendbarkeit

neering. Die einzelnen Systementwicklungsphasen werden durch spezielle Ergonomiebetrachtungen, so genannte „Prozessschritte“ oder „Usability Tasks“, angereichert. Das derzeitige Referenzmodell ist in Abbildung 1 dargestellt.

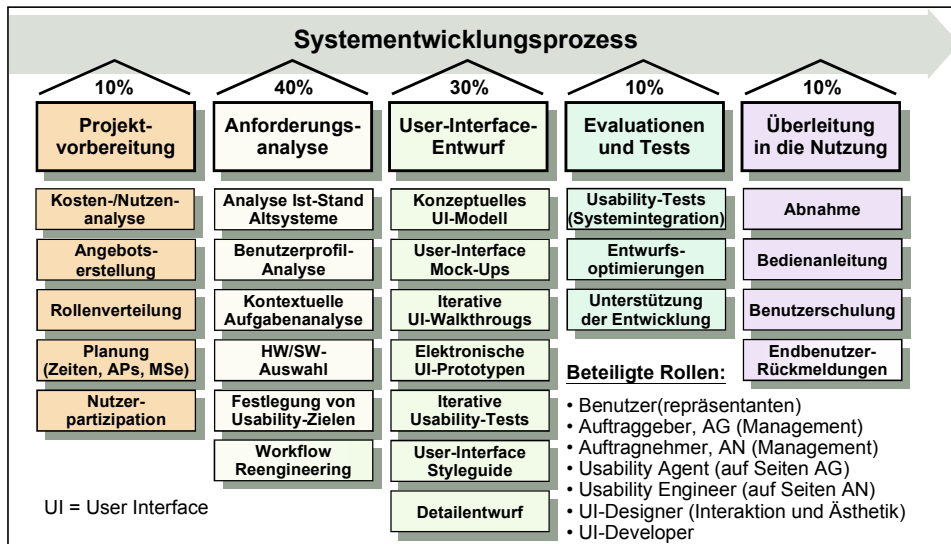


Abb. 1: Ein Referenzmodell für Usability Engineering

3 Unterstützungswerkzeuge

Auf dem Markt sind heute über 130 Werkzeuge³ (Programmierbibliotheken, User Interface Management Systems, User Interface Builder, Icon Builder, Application Frameworks, Toolkits) bekannt (Myers, 1997), die alle den Zweck verfolgen, den Programmierer bei der Gestaltung und Programmierung der Bedienschnittstelle zu unterstützen bzw. in einigen Fällen ihm die Programmierarbeit ganz abzunehmen.

Aber die Gestaltung der Bedienschnittstelle beschränkt sich nicht, wie oft fälschlicherweise angenommen, auf das reine Programmieren, sondern ist in den komplexen Prozess des Usability Engineering eingebunden. Praktisch keines dieser 130 Werkzeuge vermittelt das nötige Wissen und unterstützt Entwickler und Programmierer in allen wichtigen Phasen dieses Usability Engineering Prozesses. In der Literatur wird die Notwendigkeit von Tools beschrieben, die in Anlehnung an CASE-Tools mit dem Begriff CAUSE-Tools (Computer Aided Usability Engineering) eingeführt wurden. Nielsen (1997) kommt in seinem Buch "Usability Engineering" zu dem Schluss: „There are multiple tasks in the usability engineering lifecycle that could be performed more efficiently with computerized tools, there are almost no such tools commercially available...“.

Neben solchen mehr allgemeinen Betrachtungen von Vorgehensweisen gibt es auch eine Reihe von konkreteren Vorgehensmodellen. Ein Beispiel ist *TASK* (Beck & Janssen 1993: Technik der aufgaben- und benutzerangemessenen Software-Konstruktion). Hier wird der Forderung

3 Zusammenfassende Darstellung auf der Webseite der CMU Carnegie Mellon University, Pittsburgh, <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/bam/www/toolnames.html>

nach stärkerer Aufgabenorientierung und angemessener Beteiligung der Benutzer Rechnung getragen. Ferner werden soziale, organisatorische und technische Anforderungen berücksichtigt.

Ein elektronisches Unterstützungs- und Informationssystem sollte also neben der eigentlichen Aufgabe des Softwareentwurfs ebenso all jene Bereiche abdecken, die damit mittelbar oder unmittelbar verbunden sind. In der Software-Entwicklung kommt solchen so genannten Sekundäraufgaben (Ulich 1998) eine besondere Bedeutung zu. Durch ein hohes Maß an Aufgabenunsicherheit und Aufgabeninterdependenz besteht gerade hier ein größerer Aufwand an Planung, Koordination und projektinterner Kommunikation (Brodbeck 1996). Die im Rahmen des Usability Engineering verstärkte Benutzerbeteiligung kann zudem die Bewältigung zusätzlicher Sekundäraufgaben erforderlich machen und den Entwicklungsprozess erschweren und verzögern (Selig 1986). Diesen Schwierigkeiten kann durch eine Förderung von Kooperation und Kommunikation über die Bereitstellung von adäquaten Unterstützungswerkzeugen entgegengewirkt werden.

4 Fragestellung

Zur Gestaltung eines Unterstützungs- und Informationssystems, das eine effektive Begleitung des Entwicklungsprozesses gewährleisten soll, ist eine genaue Kenntnis der Anforderungen zukünftiger Benutzer notwendig. Wandke u. a. (1999) konnten in einer 1998 durchgeführten Untersuchung zeigen, dass das Wissensmanagement einen hohen Stellenwert bei der Softwareentwicklung einnimmt. Anliegen der von uns durchgeführten Untersuchung ist einerseits die Bewertung konkreter Bestandteile eines möglichen Unterstützungssystems und deren Einordnung in die Phasen der Systementwicklung. Zum anderen interessierte uns die Differenzierung der Unterstützungswünsche in unterschiedlich strukturierten Entwicklerteams mit variierenden Aufgabenverteilungen. Dieser Aspekt wird jedoch im folgenden Beitrag nicht differenziert dargestellt. Folgende Fragen wollen wir mit unserem Beitrag beantworten:

- Welche Probleme treten während des Entwicklungsprozesses besonders häufig auf?
- Welche Elemente sollte ein elektronisches Unterstützungs- und Informationssystem enthalten? Welche Unterstützungswerkzeuge würden in einem solchen System besonders favorisiert werden?
- Welche Rolle spielen die verschiedenen Unterstützungsbereiche in den einzelnen Phasen des Systementwicklungsprozesses?

5 Methodik und Durchführung

5.1 Stichprobe

In die Untersuchung wurden 16 Personen aus Entwicklerteams vier verschiedener Unternehmen einbezogen, von denen acht als Softwareentwickler, vier als Projektleiter und drei weitere in den Bereichen Konzeption, Produktdesign, Hardwarebetreuung und Usability Testing tätig sind. Die Teams waren durch räumliche Nähe gekennzeichnet und wiesen unterschiedliche Funktions- und Aufgabenteilungen auf. Die Teilnehmer waren zwischen 25 und 40 Jahre alt und verfügten im Mittel über eine gut siebenjährige Berufserfahrung in der Systementwicklung.

5.2 Befragungsinstrumente und Durchführung

Die Befragungen wurden als Einzelinterviews durchgeführt. Zu Beginn füllten die Teilnehmer einen kurzen Fragebogen aus, der Fragen zu biographischen Daten, zur Funktion und zu den typischen Arbeitsaufgaben enthielt. Anschließend wurde ein halbstrukturiertes Interview durchgeführt, dem eine szenariobasierte Befragungstechnik zugrunde lag. Die Entwickler sollten sich

dazu an ein für sie typisches Projekt erinnern und den Ablauf des Entwicklungsprozesses differenziert schildern. Dabei waren die einzelnen Arbeitsaufgaben und die damit verbundenen Kooperationen zu beschreiben und auftretende Probleme zu benennen. Außerdem sollten die Befragten angeben, woher sie sich notwendige Informationen und Werkzeuge besorgt haben und welche weitere Unterstützung sie sich zur Erleichterung der Aufgabenbewältigung gewünscht hätten.

Den zweiten Teil der Befragung bildete die skalierte Bewertung möglicher Bestandteile eines elektronischen Unterstützungs- und Informationssystems hinsichtlich ihrer Bedeutung im Entwicklungsprozess. Dazu wurde den Teilnehmern eine computerbasierte Liste mit 54 nach Unterstützungsbereichen sortierten Informationen und Werkzeugen vorgelegt (Tabelle 1).

Tab. 1: Klassifikation des Fragebogens nach Unterstützungsbereichen

Unterstützungsbereich	Beispiele aus der Bewertungsliste
Projektmanagement	ToDo-Listen, gemeinsamer Terminkalender, Meilensteinplanung
Projektinformation	Projektdokumentation, Dokumentation zu Usability-Tests
Mitarbeiterinformation	Kollegensteckbriefe, Aufgabenübersicht
Kommunikation	Chat, Diskussionsforum, Newsletter, Netmeeting
Ergonomiewissen	Ergonomie-Handbuch, Gestaltungsregeln, Vorgehensmodelle
Erfahrungswissen	Erfahrungsberichte, Projektdatenbanken, Standard-Dialoge
Allgemeine Informationen	Produktbeschreibungen, nützliche URLs
Methoden und Tools	Entwicklungsumgebungen, Evaluationstools, Checklisten

Einerseits sollten die Befragten einschätzen, für wie wichtig sie diese Unterstützungselemente halten, andererseits ihre Bedeutung für die einzelnen Phasen des Entwicklungsprozesses beurteilen, wobei zur Phaseneinteilung das in Abschnitt 2 dargestellte Referenzmodell des Systementwicklungsprozesses herangezogen wurde (vgl. Abbildung 1). Die Liste konnte außerdem um nicht genannte Elemente ergänzt werden. Um den Teilnehmern die Möglichkeit zu geben, ihre Prioritäten beim Unterstützungsbedarf noch einmal hervorzuheben, wurden sie zum Abschluss gebeten, ihre drei wichtigsten Unterstützungswünsche schriftlich zu formulieren.

Die Befragungen dauerten durchschnittlich zwei Stunden und wurden von einem Interviewer durchgeführt. Die Aussagen aus dem halbstrukturierten Interview wurden von einer zweiten Person in einem vorstrukturierten Protokollbogen festgehalten.

6 Ergebnisse

6.1 Probleme im Entwicklungsprozess

Die meisten Probleme, die von den Entwicklern in den Interviews benannt werden, betreffen organisatorische Unzulänglichkeiten im Entwicklungsprozess. Von den 16 Befragten geben 13 Probleme an, die das Projektmanagement betreffen. Häufig wurden fehlende Vorgaben hinsichtlich der Anforderungen des zu entwickelnden Systems bemängelt. Die unklaren Anforderungen führten oft zu Verzögerungen, da nachträglich Funktionalitäten eingearbeitet bzw. neue Entwürfe erarbeitet werden mussten. In diesem Zusammenhang kritisieren auch immerhin noch die Hälfte der Teilnehmer eine mangelnde oder zähe Kommunikation, ungenügende Absprachen zwischen dem Management und den Entwicklern sowie Kooperationsprobleme mit abhängigen Teilprojekten und Unterbeauftragungen. Des weiteren wünschen sich etwa 2/3 der Befragten eine sorgfältigere Dokumentation während des Entwicklungsprozesses und einen leichteren Zugriff auf diesbezügliche Ressourcen. Dies betrifft sowohl Informationen über die aktuelle Pro-

jektplanung und -koordination sowie die Dokumentation der einzelnen Phasen des Entwicklungsprozesses als auch Erfahrungen aus vorangegangenen Projekten.

6.2 Unterstützungswünsche

Insgesamt sind die im zweiten Teil der Befragung vorgelegten Unterstützungsmöglichkeiten von den Teilnehmern der Untersuchung als durchweg positiv eingeschätzt worden. Fast 90 % der möglichen Bestandteile eines elektronischen Unterstützungs- und Informationssystems wurden als eher wichtig oder sehr wichtig beurteilt. Beim Vergleich der a priori festgelegten Unterstützungsbereiche, denen die möglichen Informationen und Werkzeuge zugeordnet wurden, zeigt sich jedoch ein signifikanter Effekt auf die Bewertung der Wichtigkeit für den Entwicklungsprozess [$F(7,9) = 6.91, p < 0.01$]. Am wichtigsten sind den Befragten dabei unterstützende Werkzeuge und Informationen in den Bereichen Projektmanagement und Projektinformation (Abbildung 2).

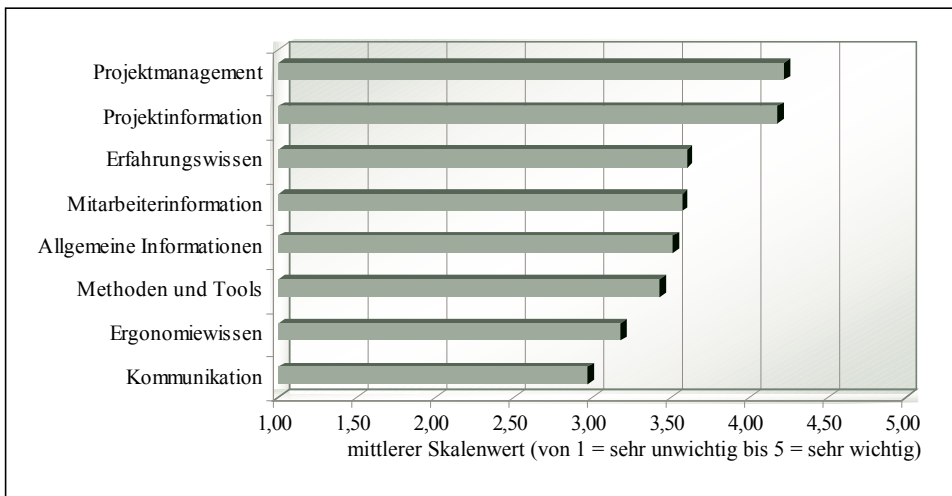


Abb. 2: Bewertung der Unterstützungselemente nach Unterstützungsbereichen

So finden sich unter den am stärksten favorisierten Elementen eines elektronischen Unterstützungs- und Informationssystems neben Programmierwerkzeugen auch Tools zum Zugriff auf Projektdokumentation, speziell der Entwurfsphase und ToDo-Listen. Werkzeuge zur Kommunikation wie Chat, Newsletter, Diskussionsforen und CSCW-Systeme werden hingegen als weniger bedeutsam angesehen (Tabelle 2). In den Interviews wurde allerdings deutlich, dass durch die enge räumliche Zusammenarbeit der Projektmitarbeiter das Gros des fachlichen Austausches informell über direkte Kontakte erfolgt und daher hier eine elektronische Unterstützung von geringerer Priorität ist. Die Möglichkeit des E-Mail-Versendens, die wegen ihrer hohen Selbstverständlichkeit sehr weit oben rangiert, stellt hierbei eine Ausnahme dar. Informationen hingegen, die direkt die projektinterne Organisation und Kooperation betreffen, werden in stärker formalisierter und aufbereiteter Weise gewünscht.

Ein Einfluss der Zugehörigkeit zu einer der vier Projektgruppen lässt sich bei der Bewertung der einzelnen Unterstützungsbereiche nicht zeigen [$F(3,12) = 0.65, n. s.$].

Tab. 2: Rangreihe der 5 wichtigsten und 5 unwichtigsten Elemente [1 = sehr unwichtig bis 5 = sehr wichtig]

Rang	Werkzeug	Wichtigkeit
1.	Projektdokumentation	4.94
2.	Entwicklungsumgebungen	4.79
3.	E-Mail	4.75
4.	Dokumentation zum User-Interface-Entwurf	4.73
5.	ToDo-Listen	4.63
:	:	:
:	:	:
50.	Schulungseinheiten, Lernprogramme, CBT	3.25
51.	Infoboard (elektronische Pinnwand), Newsletter	3.00
52.	Diskussionsforum	2.80
53.	CSCW (z. B. Application Sharing, Netmeeting)	2.77
54.	Chat	2.07

Der Wunsch nach klaren Vorgaben und einer gut gepflegten Dokumentation spiegelt sich auch in den Listen, in denen die Untersuchungsteilnehmer ihre drei wichtigsten Wünsche angeben sollten, wider. Die überwiegende Mehrheit (12 von 16) nennen unter den drei dringlichsten Unterstützungswünschen eine bessere Projektkoordination bzw. mehr Informationen und bessere Tools zu Projektmanagement und -dokumentation. Außerdem äußern gut die Hälfte der Befragten Wünsche, die sich auf den Zugriff von Erfahrungswissen bzw. die eigene Weiterbildung beziehen.

Aus den Interviews ging auch hervor, dass der Wissensabruf *on demand* gegenüber organisierten Schulungseinheiten bevorzugt wird, was den niedrigen Rangplatz dieser Angebote erklären könnte. Betrachtet man nur die Elemente, die sich in die Bereiche Erfahrungs- und Ergonomiewissen einordnen lassen, zeigt sich außerdem, dass konkrete und direkte Informationen und Erfahrungen, die einen schnellen Transfer auf aktuelle Probleme ermöglichen, am stärksten favorisiert werden (Tabelle 3).

Tab. 3: Rangplätze der Elemente zu Ergonomie-, Erfahrungswissen [1 = sehr unwichtig bis 5 = sehr wichtig]

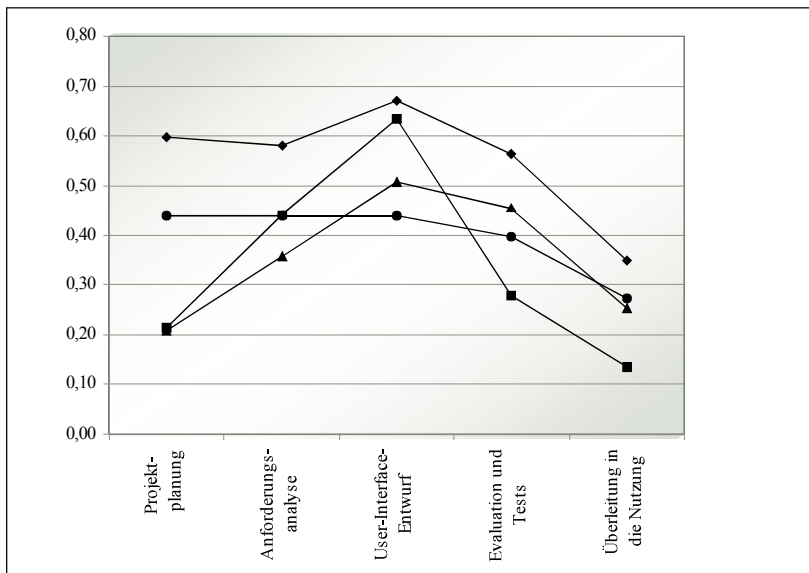
Rang	Werkzeug	Wichtigkeit
6.	User-Interface-Styleguides	4,54
9.	Erfahrungs- und Ergebnisberichte	4,50
10.	Prototypen	4,47
12.	Problemlisten	4,38
18.	Mock-Ups erfolgreicher Oberflächen	4,21
19.	Fehlerdatenbank	4,21
23.	Handbuch der Ergonomie	4,15
24.	Standard-Dialoge und -Module	4,15
25.	Normen und Standards	4,14
26.	Wissensdatenbank	4,13
29.	Gestaltungsregeln	4,08
30.	Gut-Schlecht-Beispiele	4,08
33.	Projektdatenbanken	4,00
40.	Expertennetzwerk	3,77
41.	Vorgehensmodelle	3,77
47.	Case-based Reasoning	3,55
49.	Schulungseinheiten, Lernprogramme, CBT	3,25

6.3 Unterstützungsbedarf und die Phasen des Entwicklungsprozesses

Zur Analyse des Phasenverlaufs der Unterstützungswünsche wurden die in Tabelle 1 im Abschnitt 5.2 dargestellten Bereiche stärker aggregiert. Die Bereiche Projektmanagement, Projektinformation und Mitarbeiterinformation bilden nun den Bereich „Projektmanagement und -dokumentation“, die Bereiche Erfahrungs-, Ergonomiewissen und Allgemeine Informationen wurden zum Bereich „Wissen“ zusammengefasst.

Beim Vergleich der einzelnen Phasen des Entwicklungsprozesses hinsichtlich der verschiedenen Unterstützungsbereiche zeigt sich sowohl ein signifikanter Haupteffekt der Phasen [$F(4,46) = 20.27, p < 0.01$] als auch eine bedeutsame Wechselwirkung zwischen den Phasen und den Unterstützungsbereichen [$F(12,144) = 2.79, p < 0.01$]. Die Unterstützungswünsche der Befragten unterscheiden sich also sowohl in Bezug auf die Phasen des Entwicklungsprozesses als auch hinsichtlich der Unterstützungsbereiche in Abhängigkeit von der aktuellen Phase (Abbildung 3).

Den größten Unterstützungsbedarf haben die Entwickler naturgemäß in der eigentlichen Entwurfsphase. In dieser Phase wird der Zugriff auf Werkzeuge und Informationen, die Ergonomie- und Erfahrungswissen betreffen, als besonders wichtig eingeschätzt. Für die Phase der Projektplanung spielen hingegen neben den über alle Phasen hinweg favorisierten Projektmanagement und -dokumentationstools auch Kommunikationswerkzeuge eine hervorragende Rolle.



Die bisher noch geringe oder fehlende Einbeziehung der Nutzer gerade in den ersten beiden Phasen des Entwicklungsprozesses und die mangelnde Verwendung vorhandenen Usability-Wissens aus zeitlichen, organisatorischen oder machtpolitischen Gründen erklärt, warum in diesen Phasen vergleichsweise wenig Unterstützungswerkzeuge der Bereiche „Wissen“ und „Methoden und Tools“ gewünscht werden. Wenn solche „Usability Tasks“ nicht realisiert werden, ist es kaum verwunderlich, dass hier ein geringerer Bedarf an entsprechenden Werkzeugen besteht. Die Ausweitung von Usability-Aktivitäten unter Einbeziehung der Entwickler auch auf diese Entwicklungsphasen könnte jedoch zu einer zunehmenden Bedeutung solcher Unterstützungstools führen.

7 Schlussfolgerungen für eine Unterstützung und ein verteiltes Wissensmanagement

Die Grundprinzipien des Usability Engineering sind in der wissenschaftlichen Welt ausreichend dokumentiert und Einzelmethoden oft auch durch praktische Erfahrungen untermauert und verfeinert. Die generelle und breite Umsetzung der Gesamtmethodik „Usability Engineering“ in der Praxis ist auf Basis der vorliegenden Erkenntnisse möglich. Der Mangel liegt in der ganzheitlichen und durchgängigen Unterstützung von Entwicklungsorganisationen und Software-Entwicklern. Eine Verfügbarmachung der Erkenntnisse in intuitiver und leicht zugänglicher Form in der gewohnten Arbeitsumgebung in Form eines Computer Aided Usability Engineering wird als ein erfolversprechender Weg betrachtet.

Den Entwicklern, Projektleitern und CASE-Teamleitern müssen wesentliche Entscheidungs- und Gestaltungsgrundlagen für die User-Interface-Entwicklung möglichst direkt in ihrer täglichen Arbeitsumgebung zur Verfügung gestellt werden. Ferner sollten im Laufe der Entwicklung gesammelte (insbesondere domänenspezifische) Erkenntnisse und Erfahrungen für Folgeprojekte verfügbar gemacht werden (Lernen durch Erfahrungen). Gerade diese konkreten Erfahrungen werden von den Entwicklern als Wissensquelle stärker geschätzt als allgemeine Modelle

und Richtlinien. Allerdings erfordern solche Elemente eine flexiblere Handhabung, da sie nicht nur den Wissensabruf sondern genauso die Wissens-bereitstellung durch die Nutzer selbst unterstützen müssen.

Es sollte nicht nur um die Unterstützung von Entwicklungsprojekten, sondern weiter gefasst auch um die Unterstützung der gesamten Organisation über Projektgrenzen hinweg gehen. Wie die Ergebnisse der Untersuchung auch zeigen, besteht neben den Unterstützungswünschen bezüglich der primären Entwicklungsaufgaben ebenso ein hoher Bedarf an Werkzeugen und Informationen, die das Projektmanagement und die Dokumentation betreffen. Diese bereitzustellen, sollte daher ebenfalls eines der vorrangigen Ziele bei der Entwicklung eines elektronischen Unterstützungs- und Informationssystems sein, um die Koordination und Kooperation in den Projekten und damit letztendlich die Qualität der Software-Entwicklung zu verbessern.

8 Literatur

- Beck, A.; Janssen, C. (1993): Vorgehen und Methoden für aufgaben- und benutzerangemessene Gestaltung von graphischen Benutzungsschnittstellen. In: Menschengerichte Software als Wettbewerbsfaktor. German Chapter of the ACM - Berichte 40. Stuttgart: B.G. Teubner
- Brodbeck, F. C. (1996): Kommunikation und Leistung in Projektarbeitsgruppen. Eine empirische Untersuchung an Software-Entwicklungsprojekten. Aachen: Shaker Verlag
- Browne, C. M. (1988): Human-Computer Interface Design Guidelines. Norwood N.J.: Ablex Publishing
- Card, S. K. ; Moran, T. P.; Newell, A. (1983): The Psychology of Human-Computer Interaction. Hillsdale, N.J.:Lawrence Erlbaum
- Constantine, L. L.; Lockwood, L. A. D. (1999): Software for Use. A Practical Guide to the Models of Us-age-Centered Design. Reading, Massachusetts: Addison-Wesley
- Mayhew, D. J. (1992): Principles and Guidelines in Software User Interface Design. NJ: Prentice Hall
- Myers, B. A. (1997): UIMSS, Toolkits, Interface Builders. In: J. Nielsen (Eds.): Handbook of User Interface Design
- Nielsen, J. (1997): Usability Engineering. London: Academic Press
- Preece, J. (1994): Human-Computer Interaction. Reading, Massachusetts, 1994: Addison-Wesley
- Schoeffel, R. (1997): Usability Engineering am Beispiel des Home Electronic System von Siemens und Bosch. In: Software-Ergonomie '97 - Usability Engineering: Integration von Mensch-Computer-Interaktion und Software-Entwicklung, German Chapter of the ACM - Berichte 49. Stuttgart: Teubner
- Shneiderman, B. (1998): Designing the User Interface. Strategies for Effective Human-Computer Interaction. Third Edition. Reading, Massachusetts: Addison-Wesley
- Selig, J. (1986): EDV-Management. Eine empirische Untersuchung der Entwicklung von Anwendungssystemen in deutschen Unternehmen. Berlin: Springer
- Smith, S. L.; Mosier, J. N. (1986): Guidelines for Designing User-Interface Software. Bedford, MA: Mitre
- Ulich, E. (1998): Arbeitspsychologie. Stuttgart: Schäffer-Poeschel
- Wandke, H.; Dubrowsky, A. & Hüttner, J. (1999): Anforderungsanalyse zur Einführung eines Unterstützungssystems bei Software-Entwicklern. in: Arend, U.; Eberleh, E. & Pitschke, K. (Hrsg.): Software-Ergonomie '99. Berichte des German Chapter of ACM Nr. 53. Stuttgart : B. G. Teubner, S. 321-334

Adressen der Autoren

Dipl.-Ing. Richard Oed
DaimlerCrysler AG
Forschung Softwaretechnologie
- FT3/SP
Postfach 2360
89013 Ulm
richard.oed@daimlerchrysler.com

PD Dr. Elke Wetzenstein / Dipl.-Psych. Anja Becker
Humboldt-Universität zu Berlin
Institut für Psychologie
Oranienburger Str. 18
10178 Berlin
wetzenstein@psychologie.hu-berlin.de
anja.becker@psychologie.hu-berlin.de