

Usability (Re-) Engineering von Legacy Systemen

Hausgemachte Unternehmenssoftware auf dem Usability-Prüfstand



Tim Schneidermeier

small worlds GbR
Bruderwöhrdstr. 15b
93055 Regensburg
tim.schneidermeier@small-worlds.de

Markus Heckner

Universität Regensburg,
Lehrstuhl für Medieninformatik
93040 Regensburg
markus.heckner@ur.de

Isabella Hastreiter

Universität Regensburg
93040 Regensburg,
isabella.hastreiter@stud.uni-regens-
burg.de

Christian Wolff

Universität Regensburg,
Lehrstuhl für Medieninformatik
93040 Regensburg,
christian.wolff@ur.de

Abstract

Unternehmenssoftware als Eigenentwicklung entwickelt sich oft über einen langen Zeitraum evolutionär weiter: Dabei ändern sich die Anforderungen an die Software laufend, der Funktionsumfang wächst kontinuierlich. Das User Interface muss entsprechend an die neuen Gegebenheiten angepasst werden und neue Eingabe- und Interaktionsmöglichkeiten für die Mitarbeiter liefern. In den wenigsten Fällen wird hierbei konsequentes Usability Engineering betrieben, das die Anforderungen der Fachebene in einem benutzbaren Navigationskonzept umsetzt. Die Folge ist häufig eine Software, die hohe Einstiegschürden für neue Benutzer bietet und auch für erfahrene Anwender deutlich effizienter sein könnte. In diesem Beitrag wird ein Projekt vorgestellt, in dem eine über zehn Jahre gewachsene Software einem Usability Reengineering unterworfen wird. Die bestehenden Prozesse wurden analysiert, ein neues Interaktionskonzept erarbeitet und durch Nutzertests evaluiert. Die dafür entwickelte Methodik und die daraus entstandenen Lessons Learned sind für vergleichbare Problemstellungen auch in anderen Projekten einsetzbar.

Keywords:

/// Usability Engineering
/// Legacy Systeme
/// Reengineering
/// Unternehmenssoftware

1. Problemstellung

Unternehmen benötigen häufig eine auf Ihre Anforderungen maßgeschneiderte enterprise resource planning-Software (ERP-Software). Auf dem Markt erhältliche Softwarepakete sind oftmals gerade für mittelständische Unternehmen zu teuer in Anschaffung und Wartung, bieten eine Vielzahl von Features, von denen ein Großteil für die Nutzung im Unternehmen irrelevant ist und entsprechen folglich eben gerade nicht den hauseigenen Anforderungen. Der Kauf einer Drittanbieter-Software von der Stange entfällt damit oft als Option. Alternativ dazu können Lösungen im eigenen Haus entwickelt werden. Diese Software-Pakete passen sich nahtlos dem täglichen Arbeitsspektrum im Unternehmen an und sollen so die Mitarbeiter bestmöglich in ihren Aufgaben unterstützen. Gerade bei mittelständischen Unternehmen werden diese Anwendungen häufig von einer kleinen IT-Abteilung entwickelt und betreut (oftmals nur 1-2 Mitarbeiter). Aus Zeitmangel und mitunter

fehlender formeller Ausbildung in Usability Engineering konzentriert man sich auf die Implementierung der notwendigen Funktionen und vernachlässigt Usability-Aspekte. Durch kontinuierliche Weiterentwicklung und Ergänzung der Softwarefunktionalität wird die Gebrauchstauglichkeit zudem eingeschränkt.

Die Benutzung solcher Software bietet folglich für die Nutzer nicht selten kein sehr positives Nutzungserlebnis: Die Navigation ist umständlich und häufig über viele Ebenen verschachtelt. Die Nutzer landen bei Ihrer Arbeit in Sackgassen, aus denen Sie mühsam zurückkehren müssen. Häufig bleibt als letzter Ausweg der Helpdesk oder der Supportmitarbeiter (vgl. Twentyman 2009).

Die Ursachen für die nicht optimale Usability von Unternehmenssoftware sind vielfältig und oft nicht eindeutig bestimmbar: Software, die über einen längeren Zeitraum verwendet und weiterentwickelt wird, muss an die Anforderungen des Unternehmens angepasst werden. Die Anzahl neuer

Features steigt so mit der Zeit stetig an, ohne dass nicht mehr benötigte Funktionen aus der Software entfernt werden. Häufig sind Funktionen nicht mehr den ursprünglichen Anforderungen zuzuordnen und die Bedeutung bestimmter Schaltflächen und Labels erschließt sich aktuellen Benutzern nicht mehr. Im Gegensatz zu Software für Endkunden sind Mitarbeiter im Unternehmen treue Nutzer: Da sie keine Alternativen haben, müssen auch wenig benutzerfreundliche Anwendungen akzeptiert und verwendet werden. Nur in wenigen Fällen wird Usability Engineering wirklich konsequent betrieben.

Die Folgen sind für Mitarbeiter und Unternehmen spürbar: Die Einarbeitung in die neue Software dauert lange, Unsicherheit vor Fehleingaben entsteht und Effizienzeinbußen stellen sich ein. Einen Ausweg aus dieser Situation bietet zum einen die Anschaffung einer Drittanbieter-Software, was in vielen Fällen an den sehr spezifischen Anforderungen scheitert. Zum anderen besteht die Möglichkeit, die bestehende Softwarelösung durch

konsequentes Usability Reengineering an die veränderten Anforderungen des Unternehmens und der Benutzer anzupassen.

Im folgenden Beitrag wird ein konkretes Projekt zur Verbesserung der Usability von einer über die Jahre gewachsenen Unternehmenssoftware vorgestellt. Neben dem Vorgehensmodell werden vor allem die konkreten Herausforderungen an das Usability (Re-) Engineering und die Lessons Learned betont.

2. Herausforderungen und Ziele

Die in Regensburg ansässige Firma R-KOM¹ bietet Privat- und Geschäftskunden Telekommunikationsprodukte an. Zum Produktportfolio gehören unter anderem Telefon-, DSL- und Glasfaseranschlüsse. Mit der auf Delphi basierten Unternehmenssoftware Zentrale Fachanwendung (ZFA) steuert die R-KOM eine Reihe von innerbetrieblichen Vorgängen: Kunden und Aufträge können in der Software erfasst, bearbeitet, umgezogen und gekündigt werden. Ebenso werden Tickets von Kundenberatern aufgenommen und von Mitarbeitern aus technischen Abteilungen bearbeitet. Die Software wurde vor ca. 10 Jahren erstmals entwickelt und seitdem laufend an die veränderlichen Anforderungen angepasst. R-KOM hat sich an small worlds mit den folgenden Anforderungen gewandt: Bestehende Funktionen sollen für die Benutzer vereinfacht und nicht (mehr) benötigte Funktionen identifiziert werden. Zudem soll ein Styleguide erarbeitet werden, mit dessen Hilfe R-KOM in Zukunft Funktionserweiterungen möglichst eigenständig und benutzerfreundlich entwerfen kann. Aus Sicht des mittelständischen Unternehmens sollten die folgenden Ziele mit diesen Maßnahmen erreicht werden: Entlastung der Mitarbeiter in Ihrer täglichen Arbeit, Effizienzsteigerung der Arbeitsabläufe und Reduktion der Aufwände für Einarbeitung und Support. Zusätzlich soll die Software als Produkt für weitere Unternehmen aus derselben Branche (mittelständische Anbieter von

Telekommunikationsdienstleistungen) angeboten und die Attraktivität der Anwendung dementsprechend für potenziell zukünftige Kunden gesteigert werden.

3. Usability Reengineering: Vorgehensmodell

Der Begriff des Reengineering hat seine Wurzeln in der Geschäftswelt und beschreibt

„... the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical modern measures of performance, such as cost, quality, service, and speed.“ (Hammer & Champy 1993)

Diese Idee aufgreifend dient als Basis für das dem Projekt zugrunde liegende Vorgehensmodell ein iteratives Usability Engineering Framework nach DIN EN ISO 9241-210 (2010). Abbildung 1 zeigt das hier angewandte Modell. [Abb. 1]

Das Projekt lässt sich in vier nicht scharf voneinander zu trennenden Phasen einteilen. In der Planungsphase wurde zunächst der Scope für den Projektumfang definiert. Nutzerbeobachtungen und ein Workshop ergaben die Anforderungen für das Usability Reengineering. Sketching (vgl. Warfel 2009) führte zu schnellen Ergebnissen für das Redesign der in den Anforderungen

erhobenen zentralen Aufgaben sowie des kompletten Look and Feel des User Interfaces. Informelle Benutzertests (vgl. Krug 2009; Nielsen 1994) wurden für unmittelbares Feedback verwendet, deren Resultate in einer neuen Iteration eingearbeitet und wieder getestet. Im nächsten Schritt wurden die Ergebnisse in einen klickbaren, interaktiven Prototypen umgesetzt und erneut getestet. Im Folgenden wird ausführlich auf die einzelnen Phasen eingegangen.

3.1. Planung

Im Kick-Off wurden Umfang und Scope des Projekts abgesteckt. Aufgrund des sehr großen Funktionsumfangs der Software wurden im ersten Schritt die wichtigsten Geschäftsprozesse für ein Re-design ausgewählt. Diese konnten vom Auftraggeber durch statistische Auswertung der laufenden Geschäftsprozesse eindeutig identifiziert werden. Die Nutzer können so in einem Großteil ihrer täglich anfallenden Arbeitsschritte unterstützt werden. Bei den so ermittelten Hauptfunktionen handelt es sich um Erfassung, Eingabe, Verwaltung und Verändern von großen Datenmengen (Personen, Verträge etc.). Um dem Projektteam einen ersten Eindruck von der Software und der Domäne zu geben, wurde unmittelbar an den Kick-Off ein Demo-Workshop abgehalten. In diesem Workshop wurden Inhalte, Funktionsumfang und bestehendes Interaktions- und

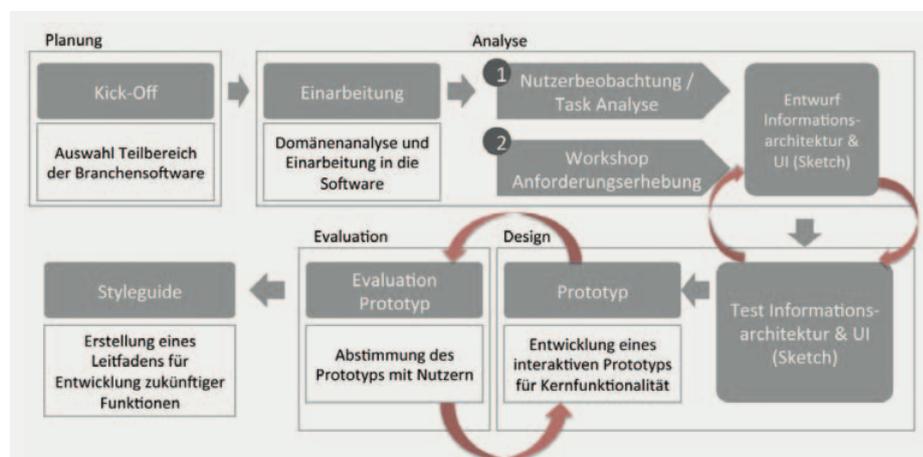


Abb. 1. Das Usability Engineering-Vorgehensmodell des Projekts.

Navigationskonzept live am Objekt demonstriert, sowie bereits von Seiten des Unternehmens erkannte Usability-Probleme benannt.

3.2. Analyse

Im nächsten Schritt erfolgte die Einarbeitung in die Anwendung und die Domäne. Ein eigener Arbeitspatz der Projektmitarbeiter beim Auftraggeber ermöglichte eine explorative Analyse der ZFA (Funktionsumfang und Informationsarchitektur). Die Methode des Cognitive Walkthrough wurde genutzt, um sich in die Arbeitsvorgänge der Mitarbeiter einzufinden und ein Gefühl für die Abläufe in der Software zu bekommen. Aufbauend darauf wurden die im Kick-Off Meeting festgelegten Hauptprozesse mit Hilfe

einer heuristischen Evaluation untersucht und Usability Probleme identifiziert. Dazu dienten Szenarien, die gemeinsam mit den Mitarbeitern entworfen wurden. Konkrete Nutzeranforderungen konnten anhand von Nutzerbeobachtungen, Contextual Inquires und (qualitativen) Interviews erhoben werden. Um unterschiedliche Nutzergruppen und ihre potenziell leicht abweichenden Anforderungen berücksichtigen zu können, wurden repräsentative Mitarbeiter mit unterschiedlichen Erfahrungsniveaus (1 Woche, 2 Jahre, 8 Jahre) ausgewählt. An drei Vormittagen wurden die Mitarbeiter von jeweils zwei Usability Experten bei ihrer Arbeit beobachtet und in kritischen Situationen Nachfragen gestellt.

Die erhobenen Anforderungen wurden dokumentiert, priorisiert und anschließend in einem gemeinsamen Workshop mit dem Entwickler, einem Vertreter der Mitarbeiter und dem Geschäftsführer vorgestellt, abgeglichen und die nächsten Schritte bestimmt. Ziele des Workshops waren:

- Unser Verständnis der bestehenden Abläufe bestätigen / abgleichen
- Konstruktive Diskussion der aktuellen ZFA-Software
- Basis schaffen für den zukünftigen Neuentwurf der Benutzeroberfläche

– Identifizierte Usability-Problembereiche präsentieren, ansprechen, diskutieren, Bewusstsein für die Relevanz von Gebrauchtauglichkeit schaffen

Zudem wurden im Workshop die durch die bisherige Analyse gefundenen Usability-Probleme aufgezeigt und mögliche Lösungen dargelegt. In einer Diskussion wurden Prioritäten aus Geschäfts- und Benutzersicht in Relation gesetzt und die Projektziele nochmals konkretisiert.

3.3. Design

Die Ergebnisse der Anforderungsanalyse wurden in einem ersten Schritt mit Hilfe von kollaborativem Sketching (vgl. Warfel 2009) zu ersten Entwürfen für die Wizards, das Dashboard sowie das komplette Look and Feel des User Interfaces umgesetzt (siehe unten Kap. 4). Das kollaborative Sketching ermöglicht ein schnelles Generieren von Ideen, die unmittelbar von den Mitgliedern des Projektteams evaluiert, und deren Ergebnisse in einer weiteren Iteration eingearbeitet werden. Gemeinsam entscheidet man sich für eine Lösung.

Das resultierende Konzept für Informationsarchitektur und den jeweiligen Screens wurde in einem nächsten Schritt unter Einsatz eines Tools (Axure) zunächst als statischer Prototyp (WYSIWYG) umgesetzt, das Konzept mittels informeller Nutzers tests evaluiert und mit dem Entwickler auf technische Machbarkeit (Delphi als Programmierumgebung für die (Weiter-) Entwicklung) überprüft. Die gewonnen Erkenntnisse wurden wiederum in den Prototypen eingepflegt und in einem auf Nutzungsszenarien² basierenden klickbaren, interaktiven Prototypen realisiert.

3.4. Evaluation

Abschließende Benutzertests mit Hilfe des interaktiven Prototypen lieferten zusätzliche Erkenntnisse, die in den finalen Prototypen eingearbeitet wurden.

Als Deliverables für den Abschluss des Projekts wurde zum einen der dokumentierte interaktive Prototyp als Spezifikation für die Benutzeroberfläche geliefert und zum anderen ein Styleguide, der die im Projekt gewonnen (Usability-) Erkenntnisse

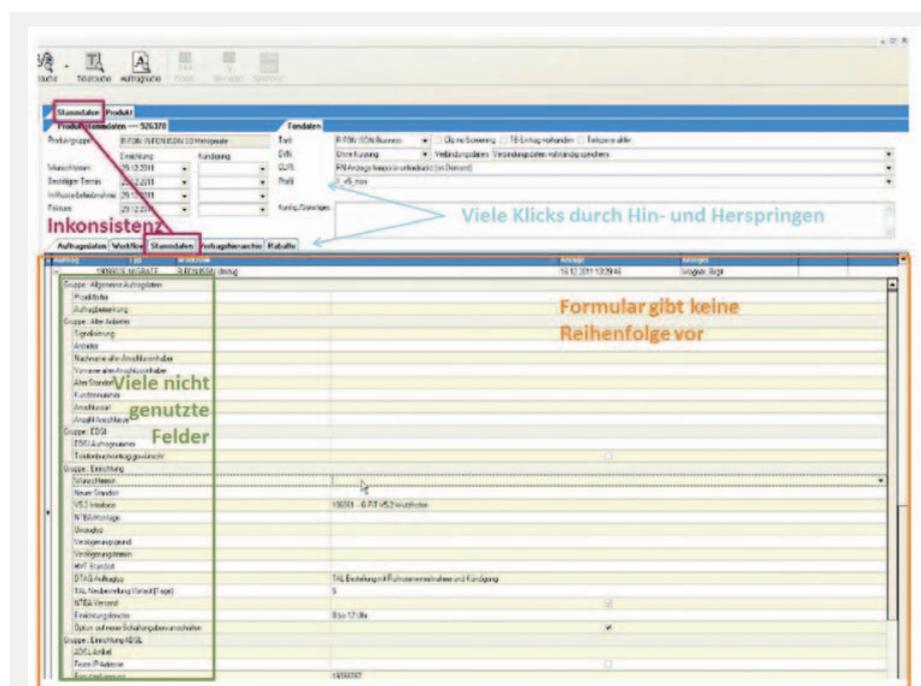


Abb. 2. Screenshot der Zentralen Fachanwendung (ZFA).

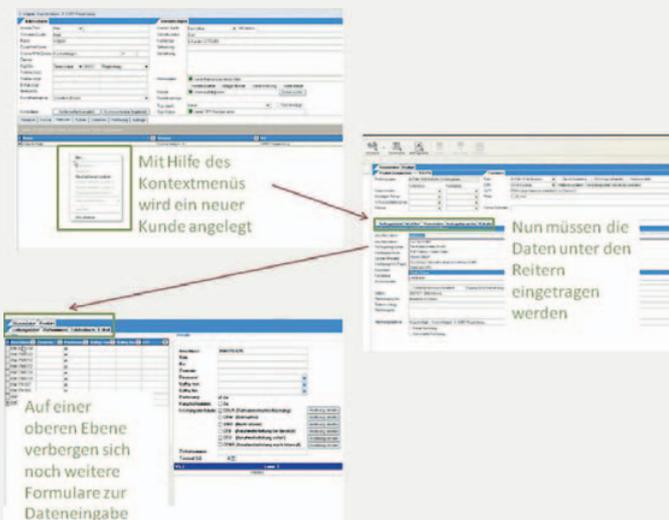


Abb. 5. Viele Reiter verursachen ein Hin- und Herspringen in der Nutzung.



Abb. 6. Prozessorientierter Wizard zur schnellen Dateneingabe.

- Zentrale Anzeige der offenen Arbeitsschritte als Strukturierungshilfe für den Arbeitsalltag
- Vereinheitlichte Suchfunktion aus Nutzersicht: Ein Suchformular durchsucht alle Datenbanken und stellt Ergebnisse gruppiert dar
- Angleich an Papierformulare zur schnelleren Datenerfassung (match between system and the real world)
- Ausblenden nicht benötigter Funktionen
- Schlüsselfunktionen über einen Klick erreichbar machen. **[Abb. 8]**
- Abbildung 8: Schlüsselfunktionen (z. B. Neuauftrag, Suchfunktion) sind stets mit einem Klick erreichbar.
- Konsistentes Layout für alle Formulare
- Kontextsensitive Hilfe zur Erläuterung von Feldnamen und Funktionen
- Umsetzung der Suche aus Nutzersicht: Ein Eingabefeld für den Suchstring liefert strukturierte Ergebnisse zurück.

6. Lessons Learned

Das Projekt konnte erfolgreich abgeschlossen werden und das überarbeitete Bedienkonzept wird vom Kunden gerade umgesetzt. Wir gehen grundsätzlich davon aus, dass die in diesem Projekt vorgefundene Mischung aus fehlendem Usability-Know-how und gewachsenen Softwarestrukturen mit hohem Funktionsumfang gerade bei kleinen und mittleren Unternehmen weit verbreitet ist. Neben der Einführung von Usability Engineering-Methoden stellt sich das konkrete Usability Re-Engineering solcher Anwendungen als relevantes Tätigkeitsfeld für Usability Consultants heraus. Folgende Punkte sind unseres Erachtens besonders relevant:

1. **Do it fast** | Durch hohe Arbeitsbelastung aufgrund parallel zu bearbeitender Projekte konnte dieses Projekt nicht so effizient bearbeitet werden, wie es prinzipiell möglich gewesen wäre. Komplexe Problemstellungen

Mitarbeiter eingegangen, erfordern aber gerade bei nicht so häufig auftretenden Use Cases immer wieder hohe kognitive Belastung. **[Abb. 5]**

5. Redesign: Wesentliche Designentscheidungen

Aus dem oben beschriebenen Prototyping und Testzyklus ergeben sich die folgenden Designentscheidungen für den Neuentwurf:

- Umsetzung eines prozessorientierten Wizards für häufige Dateneingabesequenzen. **[Abb. 6]**
- Eine auf den Unternehmenskontext angepasste Informationsarchitektur für Kunden, Produkte und deren zugehörigen Daten. **[Abb. 7]**
- Anpassen und Abgleichen der Terminologie auf Arbeitsprozesse; Reduktion der Doppelbelegungen



Abb. 7. Kunden und eindeutig zugeordnete Informationen.

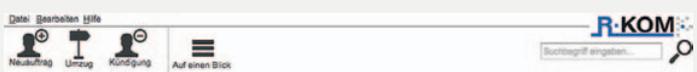


Abb. 8. Kunden und eindeutig zugeordnete Informationen.

ziehen immer wieder erneuten Einarbeitungsaufwand für die Usability-Consultants nach sich und verursachen somit eine ineffizientere Projektabwicklung durch Mehraufwände.

2. **Record your results** | Gerade bei komplexen und umfangreichen Programmen ist es für eine Usability-Analyse essentiell, dass Nutzertests, Contextual Inquiries etc. mit Video und Ton aufgezeichnet werden. Nur so lässt sich auch zu einem späteren Zeitpunkt das erfasste Problem erneut verstehen (vgl. Punkt 1). Für unsere Analyse und das Redesign war es immer wieder nötig, das Videomaterial zu sichten und erneut aus einem anderen Blickwinkel zu betrachten. Aufgrund der Komplexität ergaben sich häufig neue Perspektiven auf Probleme wie Lösungen. Die Nutzung von Morae und Annotationen mit so genannten Markern erleichtert die Analyse der Videos deutlich.

3. **Fragen, fragen, fragen** | Nicht nur bei der Anforderungsanalyse ist es essentiell, stets nachzufragen. Wenn man denkt, man habe es verstanden – nochmals nachfragen. Aus unserer Sicht können nur so Missverständnisse minimiert werden. Falsche Höflichkeit oder Angst, der Auftraggeber können den Consultant aufgrund der vielen Nachfragen für nicht fähig halten, sind völlig fehl am Platz: Für nicht fähig wird man nur eingestuft, wenn man auch nach mehrwöchiger Projektlaufzeit Dinge nicht versteht, die man zu Anfang hätte durch Nachfragen klären können.

4. **Erfolgsfaktor** | Veränderungswille beim Unternehmen ist essentiell für den Erfolg eines Usability-Projekts eines Dritten. Mit Unterstützung des Managements ist ein wichtiger Schritt für den Erfolg des Projekts getan.

5. **Geeignete Testumgebung** | Durch die räumliche Struktur des Arbeitsumfelds beim Auftraggeber blieben die Türen zum Gang hin auch während der Contextual Inquiry und den Nutzertests geöffnet. Dies legt den Verdacht nahe, dass eventuell einige kritische Äußerungen weniger deutlich geäußert wurden.

6. **Hauseigene Software** | Das sehr gute Klima im Unternehmen führte evtl. auch dazu, dass auch von Entwicklerseite her bekannte Usability-Probleme weniger deutlich formuliert und sofort relativiert wurden.

7. **Klares Abstecken von Zielen des Reengineering** | Am Anfang ließ sich der funktionale Umfang der Software auf Grund ihrer Komplexität nur schwer erkennen. Sich hier schnell einen Überblick zu verschaffen und dann mit dem Kunden die Rahmenbedingungen abklären, was im Zeitrahmen möglich ist, ist essentiell für ein gelungenes Projekt.

7. Fazit

Unternehmenssoftware zeichnet sich aufgrund des umfangreichen Funktionsumfangs häufig durch hohe Komplexität aus. Für ein zielgerichtetes und erfolgreiches Redesign ist ein Verständnis der bestehenden Prozesse deshalb essentiell. Zudem spielen die Rahmenbedingungen eine wichtige Rolle: Der Veränderungswille beim Kunden und die Unterstützung der Mitarbeiter stellten wichtige Faktoren für den Erfolg des Projekts dar.

Literatur

1. DIN EN ISO 9241-210 (2010). Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme. Beuth Verlag GmbH.
2. Hammer, M., & Champy, J. (1993). Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business.
3. Johnson, J. (2010). Designing with the mind in mind: simple guide to understanding user interface design rules. Children. Morgan Kaufmann Publishers Inc.
4. Krug, S. (2009). Rocket surgery made easy: The do-it-yourself guide to finding and fixing usability problems. New Riders.
5. Nielsen, J. (1994). "Guerrilla HCI: using discount usability engineering to penetrate the intimidation barrier." In Bias, R. G., Mayhew, D. J. (Eds.). Cost-justifying usability. Orlando, FL: Academic Press, 245-272 (<http://portal.acm.org/citation.cfm?id=186524.186639>).

6. Twentyman, J. (2009). Usability: „Lovely Software, but I can't work it“. Financial Times. (<http://www.ft.com/cms/s/0/c627386a-5a0d-11de-b687-00144feabdc0.html#axzz1pVBLkXQZ> [Zugriff 3 / 2012]).
7. Warfel, T. Z. (2009). Prototyping: A Practitioner's Guide (1st ed.). Rosenfeld Media.

¹ <http://www.r-kom.de>

² Typische Nutzungsszenarien wurden in der Analyse erhoben.