

UsER – Ein prozessorientiertes Entwicklungssystem für Usability- Engineering

Marc Paul, Amelie Roenspieß, Michael Herczeg

Institut für Multimediale und Interaktive Systeme

Zusammenfassung

Das Usability-Engineering-Repository UsER ist eine modulare, webbasierte Entwicklungsumgebung, die kollaborative Analyse, Design und Evaluation interaktiver Systeme unterstützt. Zu diesem Zweck bietet UsER diverse Module zur methodischen, dokumentarischen und kommunikativen Unterstützung an. Dieser Beitrag verdeutlicht am Beispiel von Prozessen zur Gestaltung gebrauchstauglicher interaktiver Systeme die Funktionsweise und das Zusammenspiel der einzelnen Module. Hierbei wird insbesondere die kollaborative Entwicklung durch diverse Stakeholder wie Domänenexperten, Interaktionsdesigner, Informatiker, Qualitätsmanager und nicht zuletzt Endbenutzer thematisiert und anhand funktionaler und qualitativer Eigenschaften von UsER erläutert. UsER wurde im Kontext betrieblicher Software-Entwicklung konzipiert und realisiert, in mehreren Iterationen formativ evaluiert und optimiert und befindet sich in wissenschaftlichen wie betrieblichen Kontexten im Piloteinsatz.

1 Einleitung

Die Relevanz von Usability-Engineering und damit der benutzer-, aufgaben- und kontextbasierten Entwicklung interaktiver Systeme steht inzwischen auch im betrieblichen Kontext außer Zweifel. In den konkreten betrieblichen Prozessen und Werkzeugen zeigen sich jedoch beträchtliche Unsicherheiten, Defizite und Lücken. Diese müssen durch für die Prozesse geeignete Entwicklungsplattformen geschlossen werden. Zu diesem Zweck kombinieren wir verschiedene Ansätze aus den Bereichen Software-Systems-Engineering, Requirements-Engineering und Usability-Engineering, um den gesamten Prozess der Entwicklung interaktiver Softwaresysteme zu unterstützen. Ein Ergebnis ist das Usability-Engineering-Repository UsER. Es erlaubt die flexible Kombination verschiedener Methoden und Werkzeuge, sowohl abhängig von den projektspezifischen Bedürfnissen als auch als Grundlage für die Dokumentation, Rückverfolgbarkeit und Wiederverwendung der im Entwicklungsprozess erhobenen Informationen. UsER unterstützt unter anderem den Prozess des Human-Centered Designs (HCD) (DIN EN ISO 9241-210) mit seinen sechs Grundprinzipien:

1. Das Design basiert auf einem expliziten Verständnis und Dokumentation des Benutzers, der Aufgaben und des Kontextes.
2. Benutzer und Kunden sind in den Design- und Entwicklungsprozess zu involvieren.
3. Das Design wird durch benutzerzentrierte Evaluationen getrieben und verfeinert.
4. Der Prozess ist iterativ.
5. Das Design berücksichtigt die User Experience.
6. Das Design-Team verfügt über multidisziplinäre Kompetenzen und Perspektiven.

Viele Unternehmen im Bereich der Software-Entwicklung haben die Bedeutung der Gebrauchstauglichkeit von Anwendungssoftware erkannt. Nicht immer wird jedoch ausreichend realisiert, dass Gebrauchstauglichkeit keine zusätzliche Anforderung ist, die am Ende eines Entwicklungsprozesses umgesetzt werden kann, sondern während der gesamten Entwicklung berücksichtigt werden muss. Usability-Engineering bietet eine Vielzahl an Methoden und systematischen Ansätzen, die den gesamten Entwicklungsprozess unterstützen sollten, wie beispielsweise das „Goal-Directed Design“ (Cooper, Reimann & Cronin 2007), der „Usability Engineering Lifecycle“ (Mayhew 1999) oder das bereits genannte „Human-Centered Design“. Diese Ansätze weisen – nicht ganz unerwartet – viele Ähnlichkeiten und Gemeinsamkeiten auf, die sich vor allem auf die zentrale Bedeutung von Benutzern, ihren Aufgaben und ihren Nutzungskontexten zurückführen lassen. Die konkreten Software-Entwicklungsprozesse weisen als reale betriebliche Prozesse jedoch viele Spezifika auf. Aus diesem Grund wurde UsER als offenes modulares System entwickelt, um eine flexible Anpassung an praktisch beliebige Entwicklungsprozesse zu ermöglichen.

2 Der UsER-Prozess

Um einen systematischen und berechenbaren Ansatz für den Software-Entwicklungsprozess zu unterstützen, ist UsER u.a. am abstrakten HCD-Prozess (Abbildung 1) der DIN EN ISO 9241-210 orientiert. Jede der Gestaltungsaktivitäten sollte durch eine Auswahl geeigneter Methodenmodule unterstützt werden. Die Ergebnisse dieses Prozesses werden in Form von Anforderungen in den Software-Entwicklungsprozess überführt

Jedes Modul in UsER unterstützt bestimmte Aspekte diverser Usability-Methoden. Alle entwicklungsrelevanten Informationen, modelliert als objektartige Entitäten, die mit einem der Module entwickelt werden, können wiederum mit Informationen bzw. Entitäten anderer Module verknüpft werden, wodurch semantische Beziehungen und Netze aufgebaut werden können. Das Entwicklungsprojekt selbst wird mit Unterstützung von UsER jedoch klassisch in einer linearen Struktur organisiert, wie in einem üblichen Entwicklungsdokument. Dies sichert die Abbildbarkeit auf gängige Entwicklungsprozesse. Jedes Kapitel bietet die Funktionalität eines ausgewählten Moduls. Diese können mittels Drag&Drop (Abbildung 2) in den linear organisierten Regiebereich eines jeweiligen Entwicklungsdokumentes gezogen werden. Auf diese Weise lassen sich beliebig hierarchisch strukturierte Spezifikationsdokumente erstellen. Das Entwicklungsdokument selbst wird somit zum Ausgangs- und Ankerpunkt für die Entwicklungswerkzeuge. Dies ist ein wichtiger Paradigmenwechsel, der wegführt vom text- oder tabellenzentrierten Spezifizieren und Dokumentieren, hin zu aktiven

Dokumenten, deren äußere Struktur klassisch linear ist, deren innere Struktur aber ein semantisches Netz für eine ganzheitliche Analyse, Konzeption und Evaluation darstellt.

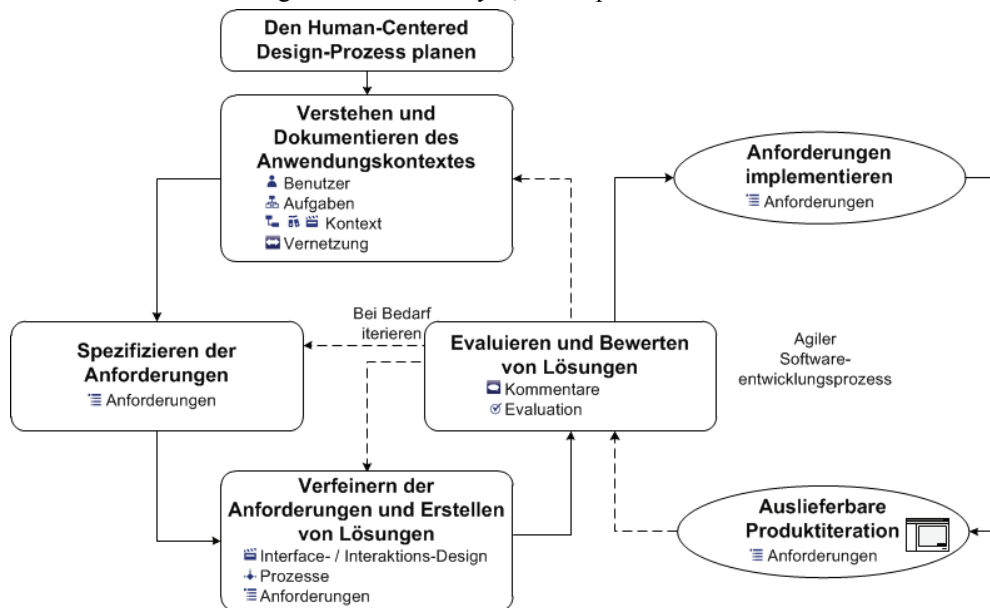


Abbildung 1: Erweiterter Human-Centered Design-Prozess

Die resultierende Struktur selbst kann gespeichert, als Dokument generiert und auch als Vorlage (Template) für zukünftige Projekte wiederverwendet werden (siehe Abbildung 2, rechts). Bisher realisierte Module unterstützen die Analyse, das Design und die Evaluation, sowie das Requirements-Management während des gesamten Software-Entwicklungsprozesses. Weitere Module aus dem Bereich sicherheitskritischer Mensch-Maschine-Systeme (Herczeg et al. 2012) oder eine Unterstützung für ein Pattern-Management werden derzeit dank der modularen Architektur mühelos integriert. Die semantischen Verweise der erhobenen Informationen helfen Entwicklungsentscheidungen aus Analyse, Design oder der Evaluation nachzuvollziehen, da diese direkt von den betreffenden Anforderungen aus zurückverfolgt werden können.

2.1 Verstehen und Dokumentieren des Anwendungskontextes

Der Iterationsprozess beginnt mit dem Verstehen und Dokumentieren des Nutzungskontextes für das zu entwickelnde System. Dabei gilt es, Benutzer und andere betroffene Interessensgruppen zu identifizieren, um anschließend deren Ziele und Randbedingungen beschreiben zu können. Um im weiteren Verlauf Lösungsansätze aus Benutzersicht evaluieren zu können, ist es notwendig, Merkmale wie Kenntnisse, Fertigkeiten, Erfahrungen, Ausbildung, Vorlieben etc. der Benutzer aufzunehmen (vgl. ISO WD 9241-230). Zu diesem Zweck dient das Benutzeranalysemodul.

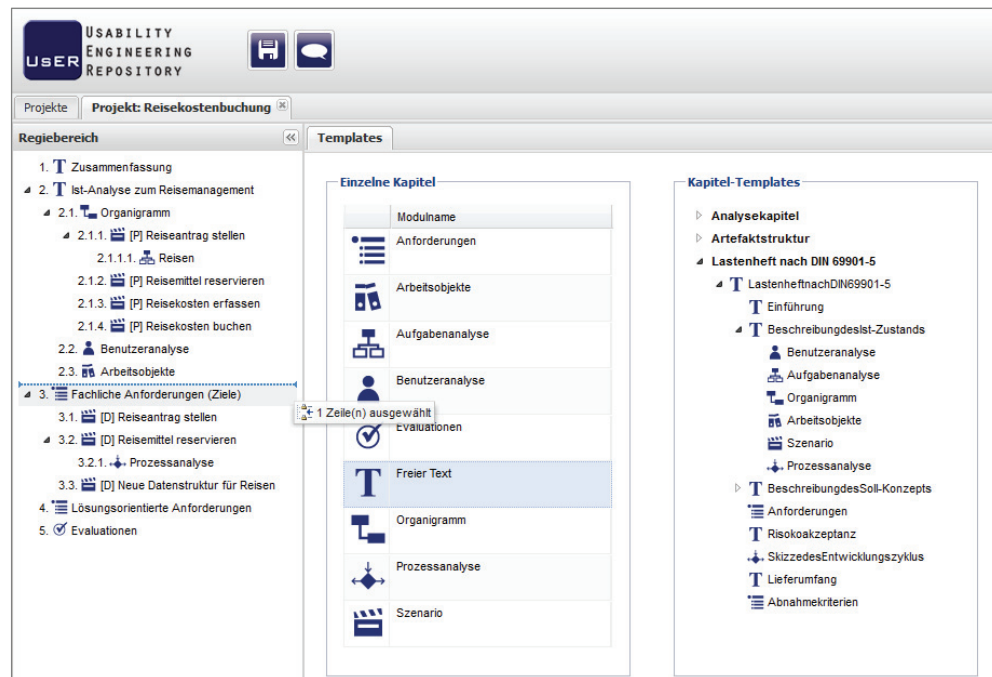


Abbildung 2: Beliebig strukturierbare Kombination von Modulen als lineares Dokument

Benutzeranalyse

Um den Kontext bzw. Anwendungsbereich einer zu entwickelnden Software zu verstehen, müssen die Betroffenen und die Akteure des Anwendungsfelds analysiert werden. Die erste Idee dieses Moduls beruhte auf dem Ansatz der Erstellung und Verwaltung von Personas (Cooper et al. 2007; Pruitt & Adlin 2006). Diese konkreten und anschaulichen Benutzermodellierungen sollten innerhalb eines Projektes in anderen Modulen referenziert und verwendet werden. In den ersten Versionen und Anwendungen dieses Moduls fiel jedoch auf, dass es für iteratives Arbeiten hilfreich wäre, Benutzer auf unterschiedlichen Abstraktionsebenen beschreiben und verfeinern zu können.

Inzwischen können im Benutzeranalysemodul neben Personas auch weniger abstrakte Stereotypen und Benutzerklassen (Pruitt & Adlin 2006; Herczeg 2009) als organisatorische Rollen oder Zielgruppen erstellt werden. So können die erhobenen Informationen einer organisatorischen Rolle genutzt werden, um sie einer Benutzerbeschreibung zu vererben und anschließend zu verfeinern. Um den Überblick zu erleichtern sowie einen Anreiz zu schaffen, möglichst detaillierte Beschreibungen von Benutzern anzufertigen, wurde ein Fortschrittsbalken zum Reifegrad der Benutzerbeschreibung integriert (Abbildung 3, rechts). Dieser errechnet sich heuristisch aus Umfang und Vollständigkeit der attributiven Beschreibungen.

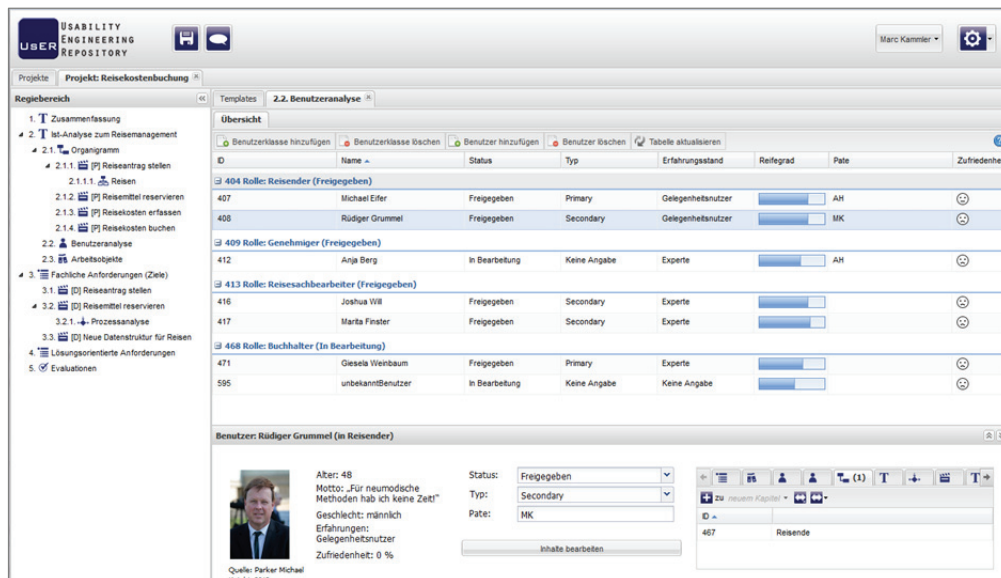


Abbildung 3: Verwaltungsansicht der Benutzerbeschreibungen

Die Verwendung von realistischen Charakteristika der Benutzer ist für die Glaubwürdigkeit und damit auch für die Akzeptanz der Benutzerbeschreibungen in der Entwicklung wichtig. Cooper propagiert daher die Verwendung sogenannter „Verhaltensvariablen“ (Cooper et al. 2007). Dies sind projektrelevante Einstellungen und Fähigkeiten von Benutzern, wie beispielsweise deren Internetnutzung. Die Antworten aus dem in das Benutzeranalysemodul integrierten Befragungsmodul werden in einer eindimensionalen Skala (Abbildung 4) dargestellt und können dann dazu verwendet werden, relevante Charakteristika des Benutzers basierend auf realen Daten zu formulieren. Auf diese Weise verbinden sich Feldforschung und Modellierung im Rahmen eines Gestaltungsprozesses (vgl. auch Mulder & Yaar 2006).

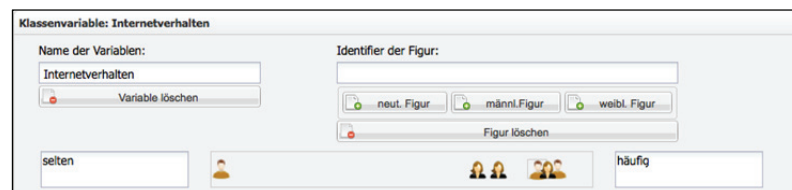


Abbildung 4: Verwendung von Benutzervariablen aus dem Fieldresearch (Felderhebungen)

Eine weitere Besonderheit des Moduls gegenüber bestehenden Systemen ist die Integration von „Benutzerzielen“. Diese können als die Motivation eines Benutzers angesehen werden, eine Aufgabe auszuführen und sollten als eine Art Fokussierung dienen, durch welche die Designer die Funktionen eines Produktes in den Blick nehmen (Cooper et al. 2007). Die Benutzerziele können über das Benutzeranalysemodul erhoben werden, wodurch sie im weiteren Entwicklungsverlauf der Benutzerbeschreibung zugeordnet werden können. Aus Projektsicht werden die Benutzerziele als Anforderungen gehandhabt. Da die von den Benutzern beschriebenen Anforderungen nur in einem bestimmten Kontext existieren, ist es notwendig,

diesen in einer geeigneten Form zu dokumentieren. Diese Aufgabe kann in UsER mit dem Szenarienmodul umgesetzt werden.

Szenarien

Die Verwendung von Szenarien hat im Usability-Engineering eine lange Tradition. Sie wurden schon früh für die Analyse von Anforderungen verwendet (Sutcliffe 1998) oder als Kommunikationsmedium bei der Entwicklung (Kyng 1995). Auch in der Anfangsphase des HCD-Prozesses eignen sie sich gut für die Beschreibung der aktuellen Arbeitsweisen. Zu diesem Zweck wurde ein eigenes Szenarienmodul in UsER entwickelt (Abbildung 5).

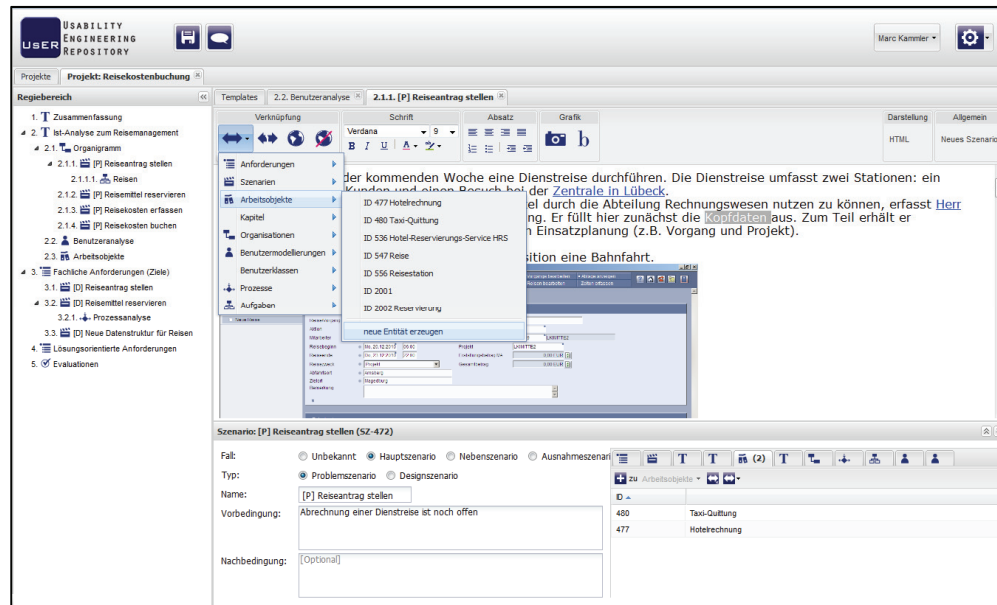


Abbildung 5: Erzeugung eines Arbeitsobjekts aus dem Szenarienmodul

Szenarien, die den gegenwärtigen Zustand eines Systems beschreiben, werden in der Literatur auch als indikative Szenarien bezeichnet (Jackson 1995). Sie fungieren als Mittler zwischen der praktizierten Realität und den konzeptuellen Modellen der Entwicklung (Pohl 2008). Sie können je nach Bedarf sowohl das Modell eines existierenden Systems als auch bei der Konkretisierung eines zukünftigen Systems (optative Szenarien) unterstützen. Bei der Modellierung eines existierenden Systems ermöglichen indikative Szenarien einen guten Ausgangspunkt für bestimmte Aspekte bzw. Perspektiven (Pohl 2008). Das Szenarienmodul von UsER erlaubt die Erzeugung und Verknüpfung projektrelevanter Entitäten wie Prozesse, Aufgaben, Arbeitsobjekte, Organisationsstrukturen und Benutzerbeschreibungen. Beispielsweise kann ein Arbeitsobjekt direkt in einem Szenario erzeugt werden. Die so erstellte Entität kann zu einem späteren Zeitpunkt in dem Arbeitsobjektmodul detaillierter beschrieben werden. Um ganze Anwendungsfälle mit einem Hauptszenario und dazugehörigen Alternativ- und Ausnahmeszenarien (Salinesi 2004) zu erfassen, wurden verschiedene Kategorisierungsattribute ergänzt. Dies ermöglicht die Gliederung der unterschiedlichen Fälle in unabhängige, aber miteinander vernetzte Szenarien. Rolland (1998) zeigt auf, wie indikative Szenarien dazu verwendet werden können, noch unerkannte Anforderungen und Benutzerziele aufzudecken. Dieses Vorgehen wird durch eine in allen Modulen integrierte Komponente

(siehe Abbildung 5, unten rechts) unterstützt, die es ermöglicht, einzelne Entitäten eines Projektes an andere Module zu übergeben.

Aufgabenanalyse

Neben generellen Informationen des aktuellen Kontextes sollen typische Arbeitsweisen der Benutzer analysiert und optimiert werden. Zu diesem Zweck ist ein genaues Verständnis von Häufigkeiten, Zeitdauer und eventueller Parallelitäten von auszuführenden Tätigkeiten notwendig. Für die Analyse von Aufgaben wurden zwei unterschiedliche Module entwickelt. Das erste ist eine Umsetzung der Hierarchical Task Analysis (HTA) nach der ursprünglichen Idee von Annett & Duncan (1967) und diverser Weiterentwicklungen (siehe z.B. Stanton et al. 2005). Bei dieser Form der Aufgabenanalyse werden Aufgaben so lange rekursiv in Teilaufgaben zerlegt, bis sie im Sinne des Feature-Driven Development (Palmer & Felsing 2002) als Features (Benutzerfunktionen) abgebildet oder auf automatisierte Prozesse (Automatisierungsfunktionen) zurückgeführt werden können. Die dabei entstehende Baumstruktur der Aufgaben kann dazu genutzt werden, ihren Knoten Szenarien zuzuordnen. Aufgaben können mit einem weiteren UsER-Modul als Aktivitätsdiagramm nach dem Standard „Business Process Model and Notation“ (BPMN) verfeinert werden. Constantine und Lockwood (1999) empfehlen die Verwendung solcher Modelle in späteren Phasen der Konzeption und Spezifikation zur Klassifizierung von Informationen. Die Aufgabenmodellierung wird im Sinne einer Hierarchical Task Analysis for Teams (HTA-T) mit den Rollen aus einer Organisationsanalyse verknüpft (Stanton et al. 2005), die im nachfolgend beschriebenen Modul erstellt wird.

Organisationsanalyse

Für die strukturierte Dokumentation betrieblicher Aufbauorganisationen werden häufig Organigramme verwendet. Das Organisationsanalysemodul ermöglicht die Erstellung solcher Diagramme. Es ermöglicht darüber hinaus, erfasste Benutzer ihren betrieblichen Organisationseinheiten oder Stellen zuzuweisen. Alle Entitäten (z.B. Aufgaben, Personas oder Szenarien) können diesen Elementen zugeordnet werden, um einen besseren Überblick und ein Verständnis der Organisationssynthese und der Arbeitsteilung zu erhalten. Erste Versuche, organisatorische Strukturen über Flow Models aus dem Contextual Design (Beyer & Holtzblatt 1997) abzubilden, wurden schnell unübersichtlich und daher verworfen. Sie können jedoch vergleichbar mit dem Prozessmodul erstellt werden.

2.2 Spezifizieren der Anforderungen

In der nächsten Gestaltungsaktivität des HCD-Prozesses geht es darum, aus den bisherigen Informationen (Benutzermerkmale, Arbeitsaufgaben und organisatorische Umgebung) und den Beschreibungen zukünftiger Arbeitsweisen mit dem System die Punkte herauszuarbeiten, die es zu ändern gilt. Die so entstehenden Anforderungen sollten neben allen produktrelevanten Forderungen auch notwendige organisatorische Änderungen oder abgeänderte Arbeitsweisen umfassen. Für die Unterstützung dieses Vorgehens wurde eine Komponente in alle Module integriert, die eine direkte Erstellung solcher Anforderungen ermöglicht (siehe Abbildung 5, unten rechts). Die auf diese Weise formulierten Anforderungen können im folgenden Modul weiter spezifiziert und bearbeitet werden.

Anforderungsmanagement

Das Anforderungsmodul bietet alle Funktionen, die für die Verwaltung und Organisation von Anforderungen erforderlich sind. Neben automatisch erfassten Informationen wie Autor, Erstellungsdatum oder Quelle (das Modul, aus dem die Anforderung stammt) können Anforderungen kategorisiert, zur Implementierung freigegeben oder deren Kosten abgeschätzt werden. Anforderungen können in beliebig viele Teilanforderungen verfeinert, sortiert und gefiltert werden. Da bei der Entwicklung dieses Moduls eine Teilmenge des Requirement Interchange Formats (Object Management Group 2011) verwendet wurde, ist auch der Export in gängige Entwicklungsumgebungen wie Eclipse möglich.

2.3 Entwerfen von Gestaltungslösungen

Dieser Schritt des HCD-Prozesses umfasst den Entwurf von Gestaltungslösungen, die die zuvor erfassten Anforderungen erfüllen. Dabei ist darauf zu achten, dass die Gestaltungslösungen die Beschreibung des Nutzungskontextes berücksichtigen (DIN EN ISO 9241-210). Mit Hilfe des bereits vorgestellten Szenarien-Moduls können diese als Design-Szenarien (Rosson & Carroll 2002) entwickelt werden, die als Diskussionsgrundlage für alle Beteiligten geeignet sind. Je nach Komplexität einer Anforderung kann bedarfsweise über eine Gestaltungslösung iteriert werden. Zur kontinuierlichen Verfeinerung der Lösungsansätze besteht die Möglichkeit, Mockups in die Szenarien zu integrieren und diese damit als Storyboards einzusetzen (Constantine & Lockwood 1999). Diese können entweder extern erstellt und via Drag&Drop in das Modul gezogen werden oder über das integrierte Mockup-Tool Balsamiq[®] erstellt werden. Um Anwendungsfälle mit einem Hauptfall und verschiedenen Alternativen und Ausnahmen (vgl. Pohl 2008) abzubilden, können Szenarien durch Links miteinander verknüpft werden.

2.4 Evaluieren und Bewerten von Lösungen

Nachdem Gestaltungslösungen entstanden sind, beginnt die Reflektion, Evaluation und Bewertung der Lösungen in Bezug auf die formulierten Anforderungen. Zwei der wichtigsten Grundsätze des HCD sind die Einbeziehung des Benutzers und die formative Evaluation von Lösungsansätzen.

Kommentare und Diskussion

Zu diesem Zweck dient eine globale Kommentarfunktion. Mit Hilfe dieser kann jedes Element eines Projektes annotiert und hinterfragt werden. Die Kommentare werden chronologisch verwaltet, wodurch Diskussionsverläufe nachvollziehbar bleiben.

Evaluationen

Das Evaluationsmodul ermöglicht die Verwendung und Erstellung von Evaluationen mit Fragebögen. Innerhalb eines Projektes kann ein Fragebogen für bestimmte Probanden freigeschaltet werden, die über einen Link in einer E-Mail an der Evaluation teilnehmen können. Die Antworten werden in einer Übersicht innerhalb des Projektes angezeigt. Im System werden bislang Fragebögen für Methoden wie SUS, ISONORM, NASA TLX und SEA bereitgestellt. Projektspezifische Fragebögen lassen sich je nach Bedarf leicht erstellen.

3 Zusammenfassung und Ausblick

Das Usability-Engineering-Repository UsER ist eine integrierte und modulare Kollaborationsplattform, die nach den Grundprinzipien eines Human-Centered Design-Prozesses verschiedene Methoden des Usability-Engineerings in flexibler Weise unterstützt. Die schon vorhandenen UsER-Module unterstützen die Analyse von Benutzern, Aufgaben, Arbeitsobjekten, Organisationsstrukturen und Arbeitsprozessen, die Konzeption von Informationsmodellen und Dialogen sowie die Evaluation von Lösungen. Sie können in einer offenen Architektur leicht um weitere Methodenmodule erweitert werden, um die Unterstützung eines benutzer-, aufgaben- und kontextgestützten Entwicklungsprozesses zu optimieren. Die Einbeziehung von Benutzern und Kunden in die webbasierte Plattform wird durch die Kommentarfunktion ermöglicht und kann darüber hinaus das iterative Vorgehen unterstützen, insbesondere bei der Rückkopplung von Evaluationsergebnissen. Die offene Architektur von UsER ermöglicht die Weiterentwicklung bestehender und die Integration weiterer Module, wodurch neue Ansätze erprobt werden können. Gebrauchstauglichkeit und Praxistauglichkeit des Systems werden seit etwa einem Jahr in wissenschaftlichen und betrieblichen Projekten erprobt und verbessert. Den Autoren ist bislang keine vergleichbar umfassende, integrierte und offene Plattform für das Usability-Engineering bekannt, die auch Schnittstellen in das Requirements- und Software-Engineering bietet und in realen betrieblichen Kontexten wirkungsvoll eingesetzt werden kann.

Literaturverzeichnis

- Annett, J., & Duncan, K. (1967). Task analysis and training design. *Occupational Psychology*, 41(July), S. 211-221.
- Beyer, H., & Holtzblatt, K. (1997). *Contextual Design. Defining Customer-Centered Systems*. San Diego: Morgan Kaufmann.
- Constantine, L. L. & Lockwood, L. A. D. (1999). *Software for Use – A Practical Guide to the Models and Methods of Usage-Centered Design*. Boston: ACM Press
- Cooper, A., Reimann, R., & Cronin, D. (2007). *About Face 3: The Essentials of Interaction Design. Information Visualization. 3. Auflage*. Indianapolis: Wiley Publishing.
- DIN EN ISO 9241-210 (2011). *Ergonomie der Mensch-System-Interaktion – Teil 210: Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme*. Berlin: Beuth.
- Herczeg, M. (2009). *Software-Ergonomie. Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme. 3. Auflage*. München: Oldenbourg.
- Herczeg M., Kammler M., Roenspieß A. (2012). Prozessorientierte Entwicklung von aufgaben- und ereignisorientierten Benutzungsschnittstellen für die Prozessführung mit Hilfe eines Usability-Engineering-Repositories (UsER). In Grandt, M & Schmerwitz, S (Eds.) Fortschrittliche Anzeigesysteme für die Fahrzeug- und Prozessführung : 54. Fachausschusssitzung Anthropotechnik der Deutschen Gesellschaft für Luft- und Raumfahrt. Koblenz: DGLR. 1-15.
- ISO WD 9241-230 (2009). User requirements for ISO standard: Human-centred design and evaluation methods. Berlin: Beuth.

- Jackson, M. (1995). *Software Requirements & Specifications: A Lexicon of Practice, Principles and Prejudices*. New York: Addison-Wesley.
- Kyng, M. (1995). Creating Context for Design. In J. M. Carroll (Hrsg.): *Scenario-based design*, S. 85-107. New York: John Wiley & Sons.
- Mayhew, D. (1999). *The usability engineering lifecycle : A Practitioner's Handbook for User Interface Design*. San Diego: Morgan Kaufmann.
- Mulder, S., & Yaar, Z. (2006). *The User Is Always Right: A Practical Guide to Creating and Using Personas for the Web*. Berkeley: New Riders.
- Object Management Group. (2011). *Requirements Interchange Format (ReqIF)*.
- Palmer, S., & Felsing, M. (2002). *A Practical Guide to Feature-Driven Development*. Prentice Hall International.
- Pohl, K. (2008). *Requirements Engineering. Grundlagen, Prinzipien, Techniken. 2. Auflage*. Heidelberg: dpunkt.
- Pruitt, J., & Adlin, T. (2006). *The Persona Lifecycle: Keeping People in Mind Throughout Product Design*. San Francisco: Morgan Kaufmann.
- Rolland, C. (1998). Guiding goal modeling using scenarios. *IEEE Transactions*, 24(12), S. 1055-1071.
- Rosson, M., & Carroll, J. (2002). Scenario-Based Design. In Jacko, J. & Sears, A. (Hrsg.): *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. 2. Auflage*, S. 1032-1050. CRC Press.
- Salinesi, C. (2004). Authoring Use Cases. In Ian, A. & Maiden, N. (Hrsg.): *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*. Chichester: John Wiley & Sons.
- Stanton, A. N., Salmon, P. M., Walker, G. H., Baber, C., & Jenkins, D. P. (2005). *Human Factors Methods: A Practical Guide for Engineering and Design*. Hampshire: Ashgate Publishing.
- Sutcliffe, A. (1998). Scenario-Based Requirement Analysis. *Requirements Engineering*, 3(1), S. 48-65.

Kontaktinformationen

Marc Paul: paul@imis.uni-luebeck.de