

# Methoden, Notationen und Werkzeuge zur Übersetzung von Anforderungen in User Interface Spezifikationen

**Thomas Memmel**  
Mensch-Computer Interaktion  
Universität Konstanz  
Universitätsstrasse 10, Box 73  
78457 Konstanz  
memmel@inf.uni-konstanz.de  
<http://hci.uni-konstanz.de>

**Thomas Geis**  
ProContext GmbH  
Deutzer Freiheit 77-79  
50679 Köln  
[thomas.geis@procontext.com](mailto:thomas.geis@procontext.com)  
[www.procontext.com](http://www.procontext.com)

**Prof. Dr. Harald Reiterer**  
Mensch-Computer Interaktion  
Universität Konstanz  
Universitätsstrasse 10, Box 73  
78457 Konstanz  
reiterer@inf.uni-konstanz.de  
<http://hci.uni-konstanz.de>

## Abstract

In diesem Workshop wollen wir unterschiedliche Methoden und Werkzeuge vorstellen und diskutieren, mit denen kreative Prozesse bei der Übersetzung von Anforderungen in benutzerfreundliche und innovative Benutzungsschnittstellen angetrieben und unterstützt werden. Dabei betrachten wir auf Text

sierende Notationsformen und Werkzeugketten ebenso wie spezielle Werkzeuge für Interaktionsdesigner. Als neue und zukunftsweisende Unterstützung von User Interface Entwicklungsprozessen stellen wir unsere Idee von interaktiven Spezifikationen vor.

## Keywords

Anforderungsanalyse, Spezifikation, Usability Engineering, User Interface Design

## 1.0 Einleitung

Der aktuelle „Chaos Report“ der Standish Group aus dem Jahr 2007 zeigt auf, dass die Softwareentwicklung auch weiterhin vor großen Herausforderungen steht. Nur 35% der IT Projekte werden erfolgreich abgeschlossen in Hinblick auf Projektlaufzeit, Budget und Umsetzung definierter Anforderungen. 46% der Projekte haben dagegen die Zeit- und Budgetvorgaben signifikant überschritten und 19% der Projekte scheitern im Durchschnitt gänzlich ohne Projektergebnis. Als primäre Ursache sind dabei besonders die frühen Phasen des Entwicklungsprozesses zu sehen, in denen die „Richtung“ für das zu entwickelnde Produkt festgelegt wird. Wer zu spät testet, muss mit in späteren Projektphasen mit teuren Problemen und Rückschlägen rechnen. Das US-amerikanische National Institute of Standards and Technology hat in einer 2002<sup>1</sup> veröffentlichten Studie erhoben, dass etwa 70% der Mängel im Entwicklungsprozess in der Phase der Anforderungserhebung entstehen. Entdeckt werden

diese meist erst während der Abnahmetests. Um einen Mangel während einer späteren Projektphase noch zu beheben muss das 50- bis 100-fache der ursprünglichen Kosten kalkuliert werden.

In diesem Workshop beleuchten wir diese Problematik in Hinblick auf die Entwicklung der Benutzungsschnittstelle (im Folgenden als User Interface, kurz „UI“ bezeichnet). Die UI Entwicklung nimmt in den meisten IT Organisationen inzwischen einen hohen Stellenwert ein und ist faktisch Teil des gesamten Entwicklungsprozesses.

Spätestens bei der UI Spezifikation und UI Visualisierung können noch übersehene oder missverstandene Anforderungen systematisch erkannt werden, wenn für die UI Spezifikation eine entsprechende Systematik zugrunde gelegt wird.

Die Spezifikation des UI kann sowohl als Teil der Anforderungsspezifikation als auch als Teil der Softwarespezifikation verstanden werden. Wie jedoch gerade dieser sensible Teil einer Softwareanwendung spezifiziert werden muss, damit Entwickler das UI anfor-

derungskonform umsetzen, wird von den meisten bekannten Usability Engineering (UE) Vorgehensmodellen offen gelassen. Doch gerade wenn der Auftraggeber eine hohe UI Qualität sicherstellen will, muss der Entwicklungsprozess eine explizite Art der UI Spezifikation vorsehen (Mommel et al., 2007a).

Wir wollen daher Methoden, Notationen und Werkzeuge diskutieren, mit deren Hilfe kreative Prozesse entlang des UI Entwicklungsprozesses bis hin zur UI Spezifikation unterstützt werden können.

## 2.0 Struktur von UI Spezifikationsprozessen

Industrielle UI Entwicklungsprojekte starten in der Regel mit einer Anforderungsspezifikation. Vor und nach der Übersetzung in ein Fachkonzept werden die Anforderungen geprüft und gegebenenfalls modifiziert. Erst nach einer Abnahme der Anforderungen und des Konzepts wird eine Systemspezifikation erzeugt. Diese ist meist die Grundlage für die Implementierung erster UI Prototypen und des finalen UI.

Anforderungen und Anforderungen sind jedoch bereits „zwei paar Schuhe“. So

<sup>1</sup> [http://www.nist.gov/public\\_affairs/releases/n2-10.htm](http://www.nist.gov/public_affairs/releases/n2-10.htm)

gibt es immer die Anforderungen aus Sicht des Kunden („Stakeholder requirements“) und Anforderungen an das System zur Umsetzung der Anforderungen des Kunden („System requirements“)<sup>2</sup>.

Eine intensive Anforderungsphase wird dann problematisch, wenn sie nicht ausreichend Spielraum für Prototyping und Evaluierung vorsieht, um so natürliche Unsicherheiten bei den Anforderungen durch das Ausprobieren passender Lösungen ausräumen zu können. Gerade in Entwicklungsprojekten, bei denen ein IT Dienstleister ab der Anforderungsspezifikation das Zepter übernimmt, sind interaktive Prototypen in der Regel vorher nicht verfügbar. In einer rein durch Spezifikation getriebenen Entwicklungskultur kann viel Zeit vergehen, bis Fachexperten und Endbenutzer zum ersten Mal das Ergebnis ihrer textuell formulierten Anforderungen ausprobieren und beurteilen können. Je größer dann die Diskrepanz zwischen Qualität und Funktionalität von Entwürfen einerseits und den Erwartungen der „Stakeholder“ andererseits ist, umso wahrscheinlicher ist es, dass die Nutzung für alle Stakeholder unbefriedigend wird.

Probleme treten bereits dann auf, wenn beteiligte Personen ihre Anforderungen nicht ausreichend gut zum Ausdruck bringen können. Viele Arbeitsprozesse laufen unbewusst ab und deren Dokumentation selbst fehlt häufig in der Anforderungserfassung. Später versuchen Experten, die lückenhaften und beliebig sortierten Anforderungen mit einem Management-Werkzeug in eine Struktur zu bringen. Das Kommunikations- und Verständnisproblem zwischen IT und Fachabteilung verstärkt sich, wenn die Projektbeteiligten sich in unterschiedlichen Sprachstilen und Abstraktionsniveaus

ausdrücken. Anforderungsspezifikationen werden dann nicht mehr verstanden und Konflikte nicht transparent.

Damit dies gelingt und letztlich auch eine vertragsbindende Anforderungsspezifikation entsteht, sind mehrere Schritte notwendig. In Abschnitt 3 betrachten wir dazu zunächst Notationen zur Erfassung und Dokumentation von Erfordernissen von Benutzern und deren Übertragung in Nutzungsanforderungen. Da es für die erfolgreiche UI Entwicklung entscheidend ist, dass Umsetzungsmöglichkeiten für erhobene Anforderungen möglichst schnell visualisiert und getestet werden können, stellen wir in Abschnitt 4 spezielle und einfach zu nutzende Werkzeuge zum UI Prototyping vor. Mit deren Hilfe soll der IT Entwicklungsorganisation der Wandel vom rein Spezifikationsgetriebenen Prototyping zur Prototyping-getriebenen Spezifikation gelingen. Dazu diskutieren wir die Möglichkeiten zu einer Verschmelzung von Anforderungsmodellen und UI Prototypen zu sogenannten interaktiven UI Spezifikationen. In diesen sehen wir die Möglichkeit, textbasierte Spezifikationsdokumente durch erlebbare Artefakte zu ersetzen. Dazu stellen wir schließlich kurz ein experimentell entwickeltes Werkzeug vor.

### 3.0 Anforderungsgetriebenes Spezifizieren von UIs

Um Anforderungen an ein UI festlegen zu können, müssen diese in Form von Nutzungsanforderungen expliziert werden. Nutzungsanforderungen sind nicht gleichzusetzen mit fachlichen Anforderungen, die primär Regeln darstellen, die ein fachlich korrekt erzielttes Arbeitsergebnis mit Hilfe des Software-Erzeugnisses / interaktiven Systems vorsehen. Nutzungsanforderungen sind auch keine „Nutzeranforderungen“. Nutzungsanforderungen

sind Anforderungen an die effiziente Nutzung eines Systems, die primär aus dem Nutzungskontext in dem das interaktive System eingesetzt wird hergeleitet werden müssen. Der Leitfaden Usability<sup>3</sup> der Deutschen Akkreditierungsstelle Technik (DATEch) liefert hierzu ein Schema zur Darlegung der Kontextdaten selbst („Kontextszenario“), das auf der Basis von Kontextinterviews mit echten Repräsentanten der Ziel-Nutzergruppe entsteht. Die Auswertung solcher Kontextszenarien in Hinblick auf implizite Erfordernisse und hieraus ableitbare Nutzungsanforderungen geschieht mit dem Schema des „ausgewerteten Kontextszenarios“. Das Schema wurde zunächst für die Herleitung von Prüfkriterien auf der Basis von DIN EN ISO 9241-11 und DIN EN ISO 9241-110 im Rahmen von Gebrauchstauglichkeitsprüfungen entwickelt, ist jedoch gleichermaßen für die Herleitung von Nutzungsanforderungen in frühen Projektphasen geeignet. Für die Nutzung des Schemas ist ein normales Textverarbeitungssystem völlig ausreichend. Die eigentliche Herausforderung liegt im methodischen Erkennen und Darlegen von impliziten Erfordernissen im Nutzungskontext. Dies erfordert einen geschulten Anforderungsanalytiker, der die Syntaxregeln für Erfordernisse und für Nutzungsanforderungen beherrscht, und die Empfehlungen der DIN EN ISO 9241-110 „Grundsätze der Dialoggestaltung“ anwenden kann.

Auf der Basis einer ausreichenden Anzahl ausgewerteter Kontextszenarien wird im Anschluss mit Hilfe eines zweiten Schemas, des „Nutzungsszenarios“ die Ausführung der zu unterstützenden Aufgaben im Nutzungskontext aus Sicht des Nutzers spezifiziert.

Diese Form der Spezifikation kann man durchaus als „Interaktionsdesign“ auf

<sup>2</sup> ISO IEC 15288 „Systems and software engineering - System life cycle processes“

<sup>3</sup> <http://www.datech.de/share/files/Leitfaden-Usability.pdf>

der Basis von Nutzungsanforderungen verstehen, die die Brücke zwischen Anforderungsspezifikation und Prototyping darstellt. Auch hierfür reicht ein Textverarbeitungssystem.

Wesentlich für den erfolgreichen Einsatz des Schemas für das Nutzungsszenario ist wiederum die Professionalität des menschlichen „Editors“, um die Interaktion zwischen Nutzer und System in Hinblick auf zu erfüllende Nutzungsanforderungen zu spezifizieren. Nutzungsanforderungen und Nutzungsszenarien, die methodisch sauber hergeleitet und dokumentiert sind, bieten eine ideale Grundlage für das Prototyping, da wichtige Fragen bereits vor dem Prototyping adressiert wurden und so „vorhersehbare Überraschungen“ beim Prototyping und Entwicklungsbegleitenden Usability Testing bereits im Vorfeld ausgeräumt werden können.

#### 4.0 Prototyping-getriebenes Spezifizieren von UI's

Nachdem Nutzungsszenarien und Nutzungsanforderungen erzeugt worden sind, können erste Prototypen erzeugt werden. Prototyping ist eine essentielle Methode im UE Prozess, um Anforderungen zu simulieren und ein UI Design bereits sehr früh im Entwicklungsprozess mit Nutzern evaluieren zu können. In Projektsituationen mit einer hohen Beteiligung von Stakeholdern, die keine IT Experten sind, muss das Erzeugen von Prototypen einfach möglich sein. Dies ist eine Voraussetzung für einen kollaborativen Prozess. Umgekehrt erzeugen innovative Prototypen, die eine magnetisierende Wirkung haben und Enthusiasmus verbreiten können, auch innovative Teams (Schrage, 1999).

Paper Prototyping reicht jedoch in vielen Situationen dazu nicht aus, da Stakeholder sehr häufig alle Ergebnisse (auch in frühen Phasen) bereits in einem „erlebbareren“ Format kennen lernen möch-

ten. Es besteht daher eine enorme Nachfrage nach Prototyping-Werkzeugen, die detaillierte Simulationen erzeugen können, aber dennoch für alle Stakeholder zu benutzen sind (Mammel et al., 2007a/b).

Wir stellen das sehr neue und im europäischen Raum noch weniger bekannte Werkzeug iRise vor. iRise Studio knüpft an die Erfahrungen der meisten Stakeholder mit Office-Anwendungen wie PowerPoint an. Dem Nutzer stehen alle gängigen UI Elemente zur Verfügung. Diese können leicht auf der „UI Zeichenfläche“ zusammengestellt werden. Auch ActiveX Komponenten wie z.B. Flash Filme, können zum Bau visuell ansprechender Prototypen integriert werden. Mit Hilfe von Templates und Master-Komponenten sind generische Änderungen leicht möglich. Durch die Verbindung mehrerer kreierter UIs entsteht ein Storyboard. Per Knopfdruck wird hieraus eine UI Simulation erzeugt, die in einem Feedback-Verfahren mit Annotationen versehen werden kann. Das iRise Datenformat erlaubt das gepackte Weiterleiten der erzeugten Prototypen an Dritte und bietet sich damit auch als nützliche Beigabe zu jeder textuell ausgearbeiteten Spezifikation an.

Zusätzlich integriert iRise auch selbst die Möglichkeit, Texte zu hinterlegen. Dadurch können einzelne UI Elemente mit den ihnen zugrunde liegenden Use Case Narrationen oder auch Anforderungen für Designentscheidungen verbunden werden. Im Unterschied zu Styleguides stellen iRise Simulationen daher eine erlebbare Dokumentation der UI Design Rationale dar. UI-Standards sowie der Look einer Anwendung werden nicht nur textuell beschrieben, sondern auch als erlebbares Artefakt bereitgestellt. Damit kann auch das Feel einer Anwendung besser weitervermittelt werden.

Die mit Prototypen erzielten Ergebnisse werden oft in Styleguides dokumentiert und treiben den weiteren UI Entwicklungsprozess an (Mayhew 1999). Styleguides stellen keine UI Spezifikationen dar, da sie in der Regel keine Kontextspezifischen Abbildungen oder Interaktionsbeschreibungen enthalten. Vor allem wenn die Implementierung eines UI durch eine dritte Partei vorgenommen wird, muss zusätzlich zum Styleguide eine Spezifikation angefertigt werden. Mayhew (1999) erläutert, dass eine solche UI Spezifikation in erster Linie aus Abbildungen besteht und nur an solchen Stellen Text enthält, an denen eine Abbildung alleine nicht ausreicht. Aus unserer Erfahrung heraus trifft dies vor allem dann zu, wenn das interaktive Verhalten eines UI im Detail beschrieben werden soll. Wir stellen im Workshop daher unsere Idee interaktiver UI Spezifikationen vor.

#### 5.0 Interaktive UI Spezifikationen

Eine interaktive UI Spezifikation verbindet die Grundidee von Styleguides mit dem Zweck einer Spezifikation. Sie speichert wichtige Designvisionen sowie -entscheidungen und dokumentiert Standards und Testergebnisse. Die textuelle Information wird durch UI Prototypen unterschiedlichen Detailgrads (low-, medium-, high-fidelity) angereichert. Zusätzlich enthält diese interaktive UI Spezifikation auch die am Beginn des Prozesses hergeleiteten Nutzungsanforderungen und -szenarien. Darauf aufbauend werden unterschiedliche (agile) Anforderungsmodelle als visuelle Notationen integriert, darunter z.B. Use Cases oder Aktivitätsdiagramme (Mammel et al., 2007c). Die interaktive UI Spezifikation, die in einem XML-basierten Format gespeichert und mit Dritten ausgetauscht werden kann, erlaubt somit den einfachen Wechsel von abstrakten Beschreibungen (Text), graphischen Nota-

tionen (Modelle) und UI Prototypen (Design).

Mit einem experimentellen Werkzeug namens INSPECTOR (Mommel et al., 2008a/b/c/d) bauen wir auf vielversprechenden Werkzeugen wie iRise auf und ergänzen diese um eine Unterstützung von Anforderungsermittlung und UI Modellierung. Durch die Integration von Text, Modellen und Design können alle für eine UI Spezifikation notwendigen Informationen mit einem innovativen Zoom-Konzept zusammengeführt werden. Dies sorgt für eine leichte Nachvollziehbarkeit und starke Transparenz sowohl hinsichtlich der Topologie des Spezifikationsraums, als auch hinsichtlich der Entwicklungsaufgabe selbst.

## 6.0 Zusammenfassung

In unserem Workshop zeigen und diskutieren wir ausgewählte Papierbasierte und interaktive UI Spezifikationstechniken. Diese unterstützen den Usability Professional bei der Identifikation von Erfordernissen und Anforderungen. Darüber hinaus helfen sie bei der Übersetzung von Anforderungen in graphische Notationen und UI Prototypen. Mit interaktiven UI Spezifikationen stellen wir eine Herangehensweise vor, die in Verbindung mit textuell ausgearbeiteten Anforderungs- und Interaktionsspezifikationen eine frühzeitige Veranschaulichung und Validierung von Lösungskonzepten fördern und so das Risiko minimieren, das wichtige textuell dargelegte Sachverhalte mangels Veranschaulichung schlichtweg übersehen werden.

## 7.0 Literaturverzeichnis

DIN EN ISO 9241-11: Anforderungen an die Gebrauchstauglichkeit; Leitsätze, Beuth Verlag, Berlin 1998.

DIN EN ISO 9241-110: Grundsätze der Dialoggestaltung, Beuth Verlag, Berlin 2006.

ISO IEC 15288: Systems and software engineering - System life cycle processes, Beuth Verlag, Berlin 2008.

Deutsche Akkreditierungsstelle Technik in der TGA GmbH (2007): Leitfadens Usability. <http://www.datech.de/share/files/Leitfadens-Usability.pdf> [01.08.2008].

Deutsche Informatik-Akademie 2003 - 2008, Seminar „Nutzungsanforderungen an Anwendungssoftware identifizieren und spezifizieren“, <http://www.dia-bonn.de/seminare/nutzungsanforderungen.html> [01.08.2008].

Deutsche Informatik-Akademie 2003 - 2008, Seminar „User Interfaces für Anwendungssoftware - Entwurf und Prototyping“, [http://www.diabonn.de/seminare/user\\_interface.html](http://www.diabonn.de/seminare/user_interface.html) [01.08.2008].

Dzida, W.; Freitag, R. (1998): Making use of scenarios for validating analysis and design, in IEEE Transactions on Software Engineering, 0098-5589, 24(1998)12, S. 1182–1196.

Dzida, W.; Hofmann, B.; Freitag, R.; Redtenbacher, W.; Baggen, R.; Zurheiden, C.; Geis, T.; Beimel, J.; Hartwig, R.; Hampen-Neteler, W.; Peters, H. (2000): Gebrauchstauglichkeit von Software. ErgoNorm: Ein Verfahren zur Konformitätsprüfung von Software auf der Grundlage von DIN EN ISO 9241 Teile 10 und 11. Bremerhaven: Wirtschaftsverlag NW.

Dzida, W.; Freitag, R. (2001): Usability Testing – The DATEch Standard, in M. Wieczorek, D. Meyerhoff (Hrsg.): Software Quality – State of the Art in Management, Testing And Tools. Berlin: Springer, 3-540-41441-X, S. 160–177.

Geis, T.; Dzida, W.; Redtenbacher, W. (2004): Specifying usability requirements and test criteria for interactive systems. Consequences for new releases of software-related standards within the ISO 9241 series. Bremerhaven: Wirtschaftsverl. NW.

Mayhew, D. J. (1999): The usability engineering lifecycle - A Practicioners Handbook for User Interface Design, San Francisco: Morgan Kaufmann.

Mommel, T.; Vanderdonck, J.; Reiterer, H. (2008a): Multi-fidelity User Interface Specifications. In Proc. of the 15th International

Workshop on the Design, Verification and Specification of Interactive Systems (DSV-IS 2008), S. 43-57. Springer: Kingston, Canada.

Mommel, T.; Geyer, F.; Rinn, J.; Reiterer, H. (2008b): Tool-Support for Interdisciplinary and Collaborative User Interface Specification. In Proc. of the IADIS International Conference on Interfaces and Human Computer Interaction (IHCI 2008), Amsterdam, Niederlande, S. 51-60

Mommel, T.; Geyer, F.; Rinn, J.; Reiterer, H. (2008c): A Zoom-Based Specification Tool for Corporate User Interface Development. In Proc. of the IADIS International Conference on Interfaces and Human Computer Interaction (IHCI 2008), Amsterdam, Niederlande, S. 368-370

Mommel, T.; Reiterer, H. (2008d): Inspector: Interactive UI Specification Tool. In Proc. of the 7th International Conference On Computer Aided Design of User Interfaces (CADUI) 2008, Albacete, Spanien, S. 161 - 174

Mommel, T.; Reiterer, H.; Ziegler, H.; Oed, R. (2007a): Visuelle Spezifikation zur Stärkung der Auftraggeberkompetenz bei der Gestaltung interaktiver Systeme. In Proceedings of 5th Workshop of the German Chapter of the Usability Professionals Association e.V., in: Kerstin Roese, Henning Brau, Frauenhofer IRB Verlag, Stuttgart, S. 99-104.

Mommel, T.; Heilig, M.; Schwarz, T.; Reiterer, H. (2007b): Visuelle Spezifikation interaktiver Softwaresysteme. In Proceedings of the 7th Mensch & Computer conference (MCI 2007, Weimar, Germany), in: Tom Gross: Mensch & Computer 2007, Oldenbourg Verlag, S. 307-310.

Mommel, T.; Gundelsweiler, F.; Reiterer, H. (2007c): Agile Human-Centered Software Engineering. In Proc. of 21st BCS HCI Group conference (HCI 2007, UK), in: Linden J. Ball, M. Angela Sasse, Corina Sas, Thomas C. Ormerod, Alan Dix, Peter Bagnall and Tom Mc Ewan: "HCI...but as we know it", British Computer Society, S. 167-175.

Schrage, M. (1999): Serious Play - How the World's Best Companies Simulate to Innovate Harvard Business School Press.

The Standish Group International (2007): CHAOS Report 2007: The Laws of CHAOS. The Standish Group International Inc, Boston.