

DER WEG ZU EINER STÄRKEREN VERZÄHNUNG VON USABILITY ENGINEERING UND SOFTWARE ENGINEERING

Kirstin Kohler
Fraunhofer IESE
Sauerwiesen 6
D-67661 Kaiserslautern
kohler@iese.fhg.de
www.iese.fhg.de

Frank Leidermann
Bluewin AG
Hardturmstrasse 3
CH-8037 Zürich
frank.leidermann@team.bluewin.com
www.bluewin.com

Andreas Birk
sd&m AG
Löffelstraße 46
D-70597 Stuttgart
andreas.birk@sdm.de
www.sdm.de

ABSTRACT

Dieser Artikel beschreibt Herausforderungen und Lösungsansätze für das Zusammenwirken von Usability Engineering (UE) und Software Engineering (SE) in der Praxis industrieller Software-Projekte. Die Praktiken sowohl des UE als auch des SE müssen in viele Projekte noch stärker eingebracht werden. Dazu müssen UE- und SE-Methoden besser miteinander verbunden werden und auch im Praxisumfeld weiter reifen.

Keywords

Usability-Engineering, Software-Engineering

1. EINLEITUNG

Die steigende Bedeutung von Usability für Software Produkte, die mit hohem Zeitdruck und unter Berücksichtigung anderer Qualitätsaspekte, wie zum Beispiel Sicherheit und Zuverlässigkeit entwickelt werden, erfordert die Anwendung von Usability Engineering (UE) und Software Engineering (SE).

Doch trotz der Notwendigkeit, Methoden beider Disziplinen im Entwicklungszyklus von Software-Projekten einzusetzen, existieren gravierende Hindernisse. Ungenügend definierte Schnittstellen zwischen Aktivitäten beider Disziplinen führen zu Problemen, die sowohl den Entwicklungszyklus als auch die Qualität der Software negativ beeinflussen. Rollen wie die des Usability-Ingenieurs müssen auf die anderen Rollen und Aufgabenfelder eines Projektes abgestimmt werden. Terminologieprobleme müssen aufgedeckt und geklärt werden.

Das folgende Kapitel gibt einen Überblick über die Herausforderungen bei der weiteren Etablierung von

UE und SE in der Praxis von Software-Projekten. Kapitel 3 stellt dem Lösungsansätze und Empfehlungen gegenüber.

2. HERAUSFORDERUNGEN

2.1 Terminologie

Terminologieprobleme sind auf den ersten Blick leicht überbrückbar. Im Alltag des Projektgeschäftes können sie aber große Missverständnisse und Schwierigkeiten bereiten. Daher ist es wichtig, mögliche Terminologieprobleme zu kennen und darauf vorbereitet zu sein.

UE und SE haben sich weitgehend unabhängig von einander entwickelt und weisen somit Terminologiekonflikte auf. Auch innerhalb jeder einzelnen Disziplin gibt es Terminologieprobleme, die sich noch verstärken können, sobald sie auf andersartige firmenspezifische Terminologien treffen. Beispiele für unterschiedlich belegte Begriffe sind: *Design, Usability, Test, Nutzer, Interface, Prototyp*.

2.2 Konstruktive Methoden

Konstruktive Methoden des UE und des SE sind jene, die primär gestaltenden Einfluss auf ein Software-System nehmen. Beispiele aus dem Bereich des UE sind Card Sorting, Storyboarding, Prototyping. Beispiele aus dem Bereich des SE sind Methoden zum Anforderungsmanagement (z.B. Volere), die Vielzahl objektorientierter Verfahren (z.B. Rational Unified Process, Catalysis, CRC Cards) und die verschiedenen Programmier-techniken (z.B. die Benutzung von Programmier-Mustern und -Idiomen, die Konzepte des Test-Driven Programming und Pairwise Programming im Extreme Programming (XP) sowie Aspect-Oriented Programming).

Die besondere Herausforderung bei der Integration von konstruktiven UE- und SE-Methoden ist, dass es sich bei diesen Methoden oft um in sich geschlossene Vorgehensweisen handelt. Zur Integration von Methoden müssen Teile von ihnen eng aufeinander abgestimmt werden. Nicht selten ist dies recht aufwändig und führt letztlich zu einer neuen eigenständigen Methode.

Es ist erlaubt digitale und Kopien in Papierform des ganzen Papers oder Teilen davon für den persönlichen Gebrauch oder zur Verwendung in Lehrveranstaltungen zu erstellen. Der Verkauf oder gewerbliche Vertrieb ist untersagt. Rückfragen sind zu stellen an den Vorstand des GC-UPA e.V. (Postfach 80 06 46, 70506 Stuttgart).

Proceedings of the
1st annual GC-UPA Track
Stuttgart, September 2003

© 2003 German Chapter of the UPA e.V.

Auch die Übertragung der Methoden in die Projektpraxis ist oft nicht trivial. Viele Methoden sind noch nicht umfassend in der Praxis erprobt. Ihr Einsatz erfordert weitere Anpassungen und Nachbesserungen während des Projektablaufes.

Exemplarisch werden im Folgenden einige konkrete Herausforderungen beim Zusammenwirken von UE- und SE-Methoden skizziert:

Es existieren keine ganzheitlichen Erhebungsmethoden, die es ermöglichen ein vollständiges Bild aller für ein Produkt relevanten nichtfunktionalen Anforderungen zu erlangen. Im UE werden zu Projektbeginn Usability-Ziele definiert, während SE-Methoden in den frühen Phasen vorwiegend Anforderungen wie Zuverlässigkeit, Wartbarkeit und Portierbarkeit erheben. Die integrative Betrachtung der nichtfunktionalen Anforderungen (d.h. Usability und alle anderen) ist notwendig, um Abhängigkeiten und Konflikte rechtzeitig zu erkennen und zu lösen. So werden in der Praxis vielfach Entscheidungen zu User-Interface-Technologien getroffen werden, bevor die nicht-funktionalen Systemanforderungen geklärt sind.

Des Weiteren existieren keine definierten Schnittstellen zwischen UE- und SE-Aktivitäten. So ist zum Beispiel oft unklar, wie Aufgaben-Beschreibungen einer UE-Aufgabenanalyse zielgerichtet in ein objektorientiertes Design überführt werden können. Ebenso gibt es noch kein weithin anerkanntes Vorgehen, welches die Übergabe von Artefakten zwischen User-Interface-Designern und User-Interface-Entwicklern unterstützt, in dem es definiert, welche Information in welcher Notation zur Spezifikation eines graphischen User Interfaces übergeben sollte.

2.3 Evaluationsmethoden

Evaluationsmethoden überprüfen die bereits vorliegenden Artefakte der Software-Entwicklung, sei es durch Verifizierung (Überprüfung gegenüber Vorgaben anderer Artefakte) oder durch Validierung (Überprüfung gegenüber den Erwartungen von Auftraggebern oder Benutzern). Beispiele im UE sind Heuristic Evaluation, Cognitive Walkthrough oder Usability Testing. Beispiele im SE sind Inspektionsmethoden und die Testverfahren wie Akzeptanztest, Regressionstests und Unittest.

Evaluationsmethoden besitzen oft geringere Komplexität als konstruktive Methoden. Sie lassen sich somit einfacher miteinander kombinieren und auch leichter in einen gegebenen Projektkontext einfügen.

Die Herausforderungen liegen hier vor allem darin, die Einsetzbarkeit und Effizienz der Methoden zu steigern. Das betrifft sowohl die Abstimmung evaluativer SE- und UE-Maßnahmen (bspw. zwischen Funktions- und Usabilitytests), als auch die Abstimmung zwischen konstruktiven und evaluativen UE- und SE-Maßnahmen aus UE und

SE (bspw. zwischen Usabilitytesting und Komponentendesign).

2.4 Werkzeugunterstützung

Die wesentliche Herausforderung für die Werkzeugunterstützung von UE und SE ist die Integration der verschiedenen Werkzeuge und die Schaffung durchgängiger, unterbrechungsfreier Toolketten. Während hier im SE zuletzt große Fortschritte gemacht worden sind, bleiben UE-Aspekte noch weitgehend ausgespart. So wäre beispielsweise die Integration von Prototypwerkzeugen und GUI-Buildern (Code-Generatoren) wünschenswert.

2.5 Organisatorische Aspekte

Die Herausforderungen bei der Projektorganisation bestehen vor allem darin, die Belange von UE und SE zu integrieren. Industrielle Projekte orientieren sich vorwiegend an SE-Erfordernissen, während organisatorische Aspekte des UE nicht wirklich berücksichtigt werden. Dies umfasst z.B. Rollenmodelle und Zuständigkeitsregelungen sowie die Gestaltung von Kommunikationsmaßnahmen (Regelkommunikation, Ermöglichung informeller Kommunikation, Eskalationswege, etc.).

So sind Prozessverbesserungen im Bereich UE in SE-gesteuerten Unternehmen deshalb schwer einzuführen, da die Ziele der einzelnen Organisationseinheiten kontraproduktiv wirken. Sollen durch Verbesserungen im UE Supportkosten gesenkt werden, so lastet die dadurch entstehende Mehrarbeit auf der Entwicklungsabteilung. Diese wird allerdings nicht anhand der Anzahl der Supportcalls bewertet, sondern deren Erfolg wird durch die Geschwindigkeit der Umsetzung von Funktionalität bestimmt [1].

Weitere Herausforderungen entstehen in der operativen Zusammenarbeit von UE und SE-Ingenieuren sowie Designer. Infolge der ungeklärten Schnittstellen zwischen Aktivitäten des UE und SE (siehe dazu Abschnitt 2.1) kommt es in der Praxis immer wieder zu Missverständnissen bzgl. den Verantwortlichkeiten, bspw. zwischen Anforderungs-Ingenieur und Usability-Ingenieur.

An der Schnittstelle zwischen Aktivitäten des UE und SE treffen häufig verschiedene Disziplinen aufeinander. So sind im UE vorwiegend Psychologen und Designer tätig, während sich mit SE Ingenieure und Informatiker beschäftigen. Dies führt nicht selten zu Konflikten, Kommunikationsproblemen und beiderseitigem Unverständnis.

Bedingt durch die personelle Situation übernehmen Software Ingenieure in der betrieblichen Praxis häufig Aufgaben, die traditionell dem UE zugeordnet sind (z.B. User Interface Design), ohne jedoch die Grundlagen dafür erlernt zu haben [2]. Dem gegenüber steht die fehlende SE-Vorbildung vieler Usability Ingenieure im Bezug auf Implementierung. Vorschläge von User Interface Designern sind unter

Umständen schwer oder gar nicht in lauffähige Software umzusetzen, da den Usability-Spezialisten das technische Verständnis fehlt, wodurch sich zeitraubende Iterationen ergeben.

Die Einbindung von Benutzern in den Software-Entwicklungszyklus wird in vielen SE-Verfahren wenig beachtet. Demgegenüber legen UE-Methoden großen Wert auf Benutzereinbindung. Sie kümmern sich aber wenig darum, dass Benutzer in der Praxis oft nur schwer verfügbar und zugänglich sind und dass die Benutzereinbindung die Anforderungen an die Projektplanung stark erhöht.

3. LÖSUNGSANSÄTZE

3.1 Methoden und Werkzeuge

Auf dem Gebiet der Objekt Orientierten Methoden sind zunehmend Bemühungen zu erkennen, die Lücke zwischen HCI und SE durch entsprechende Vorgehensmodelle und Notationen zu schließen [3],[4],[5]. Ein Beispiel sind technologie-unabhängige „Essential Use Cases“ die als methodische Ergänzung zu den aus der Objektorientierung bekannten konkreten Use Cases entwickelt wurden [6].

User Interaction Patterns stellen einen vielversprechenden Ansatz dar, um den Übergang von Usability Anforderungen in geeignete Software Architekturen zu unterstützen. Im Esprit Projekt Status wurde eine Zuordnung von Usability Zielen zu User Interaction Patterns erarbeitet [7][8].

In anderen Ansätzen aus Praxis und Forschung wurden Vorgehen erarbeitet, die die ganzheitliche Spezifikation von User Interface Anforderungen und Funktionalen Anforderungen unterstützen, in dem sie die zu beschreibenden Elemente, Notationen und Abhängigkeiten zwischen den verschiedenen Typen von Anforderungen vorgeben. Allerdings sind diese Ansätze in der industriellen Praxis bisher noch weitgehend unbekannt [9][10].

3.2 Organisatorische Aspekte

Im Gegensatz zum SE ist das UE im industriellen Bereich selten systematisch und durchgängig in die Organisation eingebettet. Für eine organisatorische Integration von SE und UE ist diese (auch als „Strategic Usability“ [11] [12] bezeichnete) Einbettung die Grundvoraussetzung.

Diese setzt wiederum in vielen Fällen voraus, dass auf Ebene der Unternehmenskultur gegen UE („Usability Myths“ [13]) abgebaut werden. Als Leitsätze für Usability-Ingenieure können dabei dienen: „Be an engineer, not an artist.“; „Cast yourself as an ally, not an enemy“ [14]. Aber auch die Zusammenstellung interdisziplinärer Teams im User Interface Design und Development oder gegenseitige Schulungen sind hier Lösungsansätze.

Bei dem in industriellen Umfeldern dringend erforderlichen Aufbau von Kompetenzen und deren

kontinuierlichen Pflege muss ein besonderes Augenmerk auf die Integration der beiden Kompetenzbereiche SE und UE gelegt werden. Mittel für Kompetenzaufbau und –pflege sind klare Kompetenzprofile für die verschiedenen Rollen, angemessene Schulungs- und Ausbildungsprogramme, Unterstützung für Training-on-the-Job, sowie vielfältige Wissensmanagement-Maßnahmen.

Das Zusammenspiel von UE und SE in der Projektpraxis lässt sich nicht durch kurze einmalige Maßnahmen erreichen. Vielmehr sind eine schrittweise Einführung und der kontinuierliche Ausbau von UE- und SE-Praktiken erforderlich. Diese kontinuierlichen Prozesse sind dauerhaft im Projekt oder der Organisation zu etablieren. Diese Aufgabe korrespondiert teilweise mit der kontinuierlichen Kompetenzpflege, geht aber auch darüber hinaus. Fragestellungen dieser Art werden im SE als kontinuierliche Prozessverbesserung behandelt [15].

Einen sehr umfassenden Ansatz stellt die ISO 18529 dar, die ein UE-Prozessmodell beschreibt, dessen organisatorische Reife durch die in der SE-Welt zu verortenden Verfahren der ISO 15504 gemessen werden soll.

4. ZUSAMMENFASSUNG

Die Herausforderungen an der Schnittstelle zwischen UE und SE sind vielfältig und bei weitem noch nicht vollständig verstanden oder gelöst. Der Workshop soll dazu beitragen diese Problematik vor allem aus dem Blickwinkel des Usability Engineering tiefer zu beleuchten und zu ergänzen, um aufbauend darauf Verbesserungspotential zu identifizieren und Synergien zwischen beiden Disziplinen zu fördern, die sowohl zur Zufriedenheit der Endkunden als auch der Designer und Entwickler (UE und SE) beitragen.

5. REFERENCES

- [1] McCoy, T., Letter from the Dark Side: Confessions of an Application Developer, interactions, 9, November 2002
- [2] Maguire, M.C. and Graham, R., "A Survey of Usability Practice and Needs in Europe", Hanson, M.A. (Hrsg.), Contemporary Ergonomics, Taylor & Francis, 1998
- [3] Constantine, L.L., and Lockwood, L.A.D., Software for Use: A Practical Guide to the Models and Methods of Usage Centered design, Addison-Wesley, 1999
- [4] Paterno, F., Towards a UML for Interactive Systems, Proceedings of EHCI2001, May 2001, Springer
- [5] Van Harmelan, M., Object Modeling and User Interface Design, Addison-Wesley, 1999
- [6] Constantine, L.L., and Lockwood, L.A.D., Structure and Style in Use Cases for User

- Interface Design, in: Van Harmelen, M. (Hrsg.), Object-Modeling and User Interface Design, Addison-Wesley, 1999
- [7] STATUS, Software Architecture that supports Usability, Esprit Project, <http://www.ls.fi.upm.es/status/index.html>
- [8] Juristo, N., Lopez, M., Moreno A., Sanchez, I.; Improving software usability through architectural patterns, Proceedings of the ICSE Workshop Bridging the Gap Between Software Engineering and Human-Computer Interaction, May 2003
- [9] Paech, B., Kohler, K., Task Driven Requirements in Object-oriented development, in Perspectives on Requirements Engineering, Kluwer Academic Publishers, to appear
- [10] Lauesen, S., Task Descriptions as Functional Requirements, IEEE Software, März/April 2003
- [11] Rosenbaum, S. et al., A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys, Proceedings of CHI 2000
- [12] Rosenbaum, S. et al., Usability in Practice: User Experience Lifecycle – Evolution and Revolution, Proceedings of CHI 2002
- [13] Knight, J., Jefsoutine, M., Relating Usability to Design Practice, European Usability Professionals Association Conference, eupa'02, London, 2002
- [14] Mayhew, D. : Strategic Development of the Usability Engineering Function, interactions, september+october 1999
- [15] Birk, A., Rombach, D. A Practical Approach to Continuous Improvement in Software Engineering, in: Meyerhoff, D.B., Wiczorek, M.J. (Hrsg.), Software Quality – State of the Art in Management, Testing, and Tools, Springer, Berlin, 2000

Referenten



Kirstin Kohler leitet die Gruppe Usability Engineering am Fraunhofer Institut für Experimentelles Software Engineering (Fh IESE) in Kaiserslautern. Zuvor war sie mehrere Jahre für Hewlett-Packard im Bereich User-Interface Entwicklung und Einführung von User-Centered-Design Prozessen tätig. Ihr Interessensschwerpunkt liegt in der Integration von Software Engineering und Usability Engineering Methoden. Sie hat Informatik mit Nebenfach Wirtschaftswissenschaften (Dipl. Inform.) und Biologie (Dipl. Biol.) studiert.



Frank Leidermann ist seit April 2003 als Usability Consultant beim Internet Service Provider Bluewin AG in Zürich tätig. Zuvor war er fünf Jahre am Institut für Technologie und Arbeit (ITA) Kaiserslautern beschäftigt (Schwerpunkte: Usability-Engineering, WebAccessibility und Design for All), unterbrochen von einem sechsmonatigen Gastaufenthalt in der Arbeitsgruppe für Human-Computer Interaction and Assistive Technologies (ICS-FORTH).



Andreas Birk ist Berater und Projektleiter bei der sd&m AG, software design & management, in Stuttgart. Seit 1993 beschäftigt er sich intensiv mit Themen des Software-Engineering, zunächst als Berater und in der angewandten Forschung. Er hat Informatik mit Nebenfach Wirtschaftswissenschaften (Dipl.-Inform.) in Kaiserslautern studiert und ist im Bereich Software Engineering promoviert (Dr.-Ing.).

USABILITY PROFESSIONALS UND REQUIREMENTS ENGINEERING

USABILITY PROFESSIONALS UND REQUIREMENTS ENGINEERING ERFAHRUNGEN UND TRENDS

Astrid Beck
GUI Design
Offenbachstr. 18-20
70195 Stuttgart
Astrid_Beck@gui-design.de
<http://www.gui-design.de>

ABSTRACT

Usability und Requirements Engineering sind als gleichrangige Aspekte im Softwareentwicklungsprozess zu berücksichtigen. Dieser Beitrag formuliert Anforderungen und Anregungen zur Umsetzung. Der Beitrag ist Motivation und Einführung zum gleichnamigen Workshop auf der Veranstaltung des GC-UPA 2003.

Keywords

Usability, Usability Professionals, Requirements Engineering, Anforderungsermittlung, SW-Entwicklungsprozess

1. EINLEITUNG

Einer meiner ersten Aufträge bestand darin, einen Style Guide für die Entwickler eines anspruchsvollen Entwicklungsprojekts zu erstellen. Ich arbeitete also einen Style Guide aus, erstellte dazu viele erläuternde Beispiele aus dem Projekt und berücksichtigte die gängigen Normen, programmierte eine Online-Hilfe und führte Workshops und Seminare zur Einführung durch. Teile des Style Guides konnten als fester Bestandteil des Fachtests etabliert werden.

Doch der Style Guide wurde quasi nicht angewandt, die Benutzungsoberflächen wurden größtenteils genauso von den Entwicklern weiterentwickelt, wie sie es schon immer gewohnt waren – nämlich nach eigenem Wissen und Geschmack. Entsprechend inkonsistent sahen die Anwendungskomponenten aus. Was lief hier falsch?

Meine Erkenntnisse möchte ich im folgenden vorstellen und sie sollen gleichzeitig den Austausch mit Usability Professionals anregen.

Es hat sich als nützlich erwiesen, in Software-

Es ist erlaubt digitale und Kopien in Papierform des ganzen Papers oder Teilen davon für den persönlichen Gebrauch oder zur Verwendung in Lehrveranstaltungen zu erstellen. Der Verkauf oder gewerbliche Vertrieb ist untersagt. Rückfragen sind zu stellen an den Vorstand des GC-UPA e.V. (Postfach 80 06 46, 70506 Stuttgart).

Proceedings of the
1st annual GC-UPA Track
Stuttgart, September 2003

© 2003 German Chapter of the UPA e.V.

Entwicklungsprojekten – wenn nötig auch wiederholt – folgende Aspekte zu diskutieren und im Projekt transparent und verbindlich zu regeln:

- ◆ Usability ist im SW-Entwicklungsprozess zu etablieren
- ◆ Usability ist Teil des Requirements Engineering
- ◆ Usabilityarbeit ist Teamwork
- ◆ Usability ist Chefsache
- ◆ Die Rolle des Usability Professionals: Screen Visagist oder Usability Consultant?

Im folgenden werden diese Aspekte genauer erläutert.

2. USABILITY IST IM SW-ENTWICKLUNGSPROZESS ZU ETABLIEREN

Wie das einführende Beispiel zeigt, ist es nicht damit getan, einen Style Guide für das Projekt zu erstellen und dann zu hoffen, dass die Anwendung von nun an benutzerfreundlicher wird.

Usability ist nicht bloß durch einen Style Guide zu bewerkstelligen, genau so wenig wie beispielsweise die Einführung eines Testtools den SW-Test verbessert oder ein QM-Handbuch die Qualität sicherstellen kann.

Usability – also eine bestens zufriedenstellende Benutzbarkeit eines Produkts – ist nicht durch Regelwerke oder einmaliges Handeln sicherzustellenden. Usability ist auch kein Selbstläufer, der einmal angestoßen sich wunschgemäß manifestieren wird. Im Gegenteil – Usability kann erst durch konsequente, wiederkehrende Prozesse etabliert werden.

Usability ist ein notwendiger Teil des Software-Entwicklungsprozesses. Dieser These werden sicher viele Usability Professionals zustimmen, im Projektalltag hat sie sich aber noch nicht überall durchgesetzt. Daher muss ständig daran gearbeitet werden. Projektverantwortliche, Teamleiter, Chefentwickler davon zu überzeugen, dass der Usability im Projekt mehr Bedeutung und Zeit beizumessen ist.

Dabei ist Usability im gesamten Softwarelebenszyklus Aufmerksamkeit zu schenken. Die benötigten Usability-Prozesse und -methoden müssen im Projektvorgehensmodell etabliert werden, dabei können als Anhaltspunkte die DIN EN ISO 13407 [2] sowie die ISO/TR 18529 [8] dienen. Gleichzeitig sorgen auf Usability zielende Aktivitäten und Meilensteine, die auch Eingang in den Projektplan finden, für Transparenz und Messbarkeit.

Usability stellt ein Qualitätsmerkmal dar, und ist im Projekt mit konstruktiven und analytischen Qualitätsmanagementverfahren sicherzustellen. Damit ist das Software-Qualitätsmanagement angesprochen und aufgefordert, mehr für Usability im Projekt und im Unternehmen zu tun.

Hartwig plädiert für ein Prozessmodell, das einen iterativen, nutzerzentrierten Ansatz verfolgt. Dabei werden die Anforderungen konsequent aus dem Nutzungskontext begründet und zu konkreten, testbaren Kriterien verfeinert [6].

Bevan und Bogomolni [1] empfehlen die folgenden zehn Methoden in den Softwareentwicklungsprozess (Planung, Analyse, Design und Test) aufzunehmen. Sie betonen, dass keine kostenintensiven Tools benötigt werden und die Einarbeitungsaufwand sich in Grenzen hält.

1. Stakeholder meeting

identify and agree on the role of usability, broadly identifying the intended context of use and usability goals, and how these relate to the business objectives and success criteria for the system

2. Context of use

collect and agree detailed information about the intended users, their tasks, and the technical and environmental constraints

3. Scenarios of use

document examples of how users are expected carry out key tasks in a specified contexts, to provide an input to design and a basis for subsequent usability testing

4. Evaluate an existing system

evaluate an earlier version or competitor system to identify usability problems and obtain measures of usability as an input to usability requirements

5. Usability requirements

establish usability requirements for the user groups and tasks identified in the context of use analysis and in the scenarios.

6. Paper prototyping

evaluation by users of quick low fidelity prototypes (using paper or other materials) to clarify requirements and enable draft interaction designs

and screen designs to be rapidly simulated and tested.

7. Style guide

identify, document and adhere to industry, corporate or project conventions for screen and page design

8. Evaluation of machine prototypes

informal usability testing with 3-5 representative users carrying out key tasks to provide rapid feedback on the usability of prototypes

9. Usability testing

formal usability testing with 8 representatives of a user group carrying out key tasks to identify any remaining usability problems and evaluate whether usability objectives have been achieved

10. Collect feedback from users

collect information from sources such as usability surveys, help lines and support services to identify any problems that should be fixed in future versions

Bei SAP verwenden Hatscher und Böringer [7] ganz ähnlich eine Mischung von Contextual Design, Goal-direct™ Design und Szenario-basiertem Design unter Mitwirkung der Kunden sowie frühzeitigem Prototyping.

Auch Strauss [10] betont die Notwendigkeit von Prototyping. Dabei hat er gute Erfahrungen vor allem mit "low fidelity prototypes", also Prototypen mit wenig aufwendigem Technikeinsatz gemacht. Darunter versteht man Papierprototypen, die manuell oder mit Grafikwerkzeugen erstellt werden. Es wird aber empfohlen, auf die Verwendung von GUI Buildern nicht zu verzichten, damit auch die technische Machbarkeit, insbesondere die Bildschirmdarstellung und -auflösung beurteilt werden kann.

Bevan und Bogomolni [1] nutzen in ihrem Projektbeispiel Papierprototypen bereits während der Aufgabenanalyse. Benutzerfunktionen wurden mit Post-It-Notes ermittelt und gruppiert.

Meiner Erfahrung nach sind es gerade die weniger techniklastigen Methoden, die bei Fachvertretern wie Softwareentwicklern gleichermaßen erfolgreich sind.

3. USABILITY IST TEIL DES REQUIREMENTS ENGINEERING

Erst die umfassende Kenntnis der zukünftigen Anwender und deren Aufgaben stellt sicher, dass das Projektkonzept für die Usability auf einer soliden Basis aufsetzen kann. Aus Sicht der Usability ist die Nutzung der späteren Produkts während der Anforderungsermittlung zu analysieren. Auf dieser Basis lassen sich dann weitere Maßnahmen wie z.B. Style Guides, Usability Tests, Benutzerzirkel erfolgreich definieren und durchführen. Neuere Ansätze des Requirements Engineering (z.B. [11]) sehen neben

funktionalen Anforderungen auch Benutzerfreundlichkeit und Anwenderanforderungen vor, aber es wird offen gelassen, wie diese zu erreichen sind und wer dies mit welchen Mitteln im Projekt dann letztendlich analysieren, entwickeln und umsetzen soll (und kann).

Usability Professionals, Software Ergonomen, Psychologen sind eher die Mahner und Besserwisser, seltener die Bessermacher. Sie wissen noch immer zu wenig vom Alltag der Softwareentwickler und agieren oftmals isoliert und nicht als Projektmitglied.

Systemanalytiker, Anforderungsermittler, Spezifikationsersteller – also jene, die Analyse durchführen und die Fachspezifikationen und Pflichtenhefte erstellen – werden unter Zeitdruck mit unvollständiger Dokumentation und widersprüchlichen Aussagen der Fachvertreter konfrontiert. Zusätzlichen Anforderungen aus Usabilitysicht wird entsprechend mit Skepsis begegnet. So hängt es stark von der "Einsicht" der Projektverantwortlichen ab, ob man sich mit Usability und deren Anforderungen beschäftigen "darf".

Projekt- oder sogar unternehmensweite Vorgaben können hier die Lösung sein, die in einheitlicher Weise festschreiben, wie Benutzeranforderungen im Rahmen der Anforderungsermittlung zu erheben und zu integrieren sind. Dazu gehört auch, die Rolle des Usabilitymitarbeiters so zu formulieren, dass dieser gleichberechtigtes Projektmitglied ist und entsprechende Aufgaben übernimmt wie z.B. Analyse, Qualitätsmanagement, Test und Evaluation (es gäbe hier noch viel aufzuzählen) und im Rahmen dieser Tätigkeit Usability umsetzt.

Ein stärkeres Augenmerk auf Requirements Engineering insgesamt zu legen, dafür plädiert Fahney in seinem Beitrag [5], da "professioneller Umgang mit Anforderungen ein Drittel bis die Hälfte des Erfolgs von erfolgreichen IT-Projekten ausmacht", Anforderungsmanagement in Projekten aber noch zu unprofessionell betrieben werde.

Kürzere Produktionslebenszyklen führen zu ständig neuen Anforderungen (s.a. Projektbeispiel, dargestellt in [5]). Dementsprechend sind die bezüglich Usability ermittelten Anforderungen nicht fix sondern ebenso einem Change-Request-Prozess zu unterwerfen. Daraus wiederum ergibt sich ein Bedarf nach kontinuierlichem Anforderungsmanagement, permanenter Zusammenarbeit mit Kunde und Benutzern, sowie ständigem Aktualisieren von Dialogoberflächen, Style Guides, Testfällen etc. (s.a. [6]).

4. USABILITYARBEIT IST TEAMWORK

Den eben geschilderten Problemen ist frühzeitig durch Rollenbeschreibungen zu begegnen. In der Sprache des Projekts ist festzulegen, wer Anforderungen ermittelt, festschreibt und freigibt und wie deren Änderungsprozess definiert ist. Bevan und

Bogomolni [1] schlagen vor, dies in einem Stakeholder meeting (s.a. Abschnitt 2) zu erarbeiten.

Benutzeranforderungen sind – beispielsweise auf Grundlage der DIN EN ISO 9241-10 – zu erheben und in die Spezifikation mit aufzunehmen. Anforderungen der verschiedenen Interessengruppen sind nicht unumstößlich, sondern ein Aushandlungsprozess. Was Usability im Projekt bedeutet, ist mit allen Teammitgliedern gemeinsam zu erarbeiten. Dies ist in der Projektdokumentation (z.B. Usability Spezifikation, Style Guide, QM- oder PM-Handbuch, Projektakte, Fachspezifikation) festzuhalten.

In dem eingangs angesprochenen Projekt (wie auch in vielen späteren Projekten) habe ich zusammen mit engagierten Entwicklern sowie Mitarbeitern der Fachseite, einem Trainer und einem Mitarbeiter des Benutzerservice ein "Style-Guide-Team" gebildet, in dem der Style Guide kritisch überarbeitet und verabschiedet wurde. Regelmäßige Treffen stellten sicher, dass aktuelle Fragestellungen mit Lösungen erarbeitet und in den Style Guide aufgenommen wurden. Die Akzeptanz des so erarbeiteten Style Guides war drastisch verbessert worden.

Um Style-Guide-Konformität auch praktisch sicherzustellen, wurden Framework-Komponenten entwickelt, die von nun an verpflichtend zu verwenden waren.

Ein Style Guide ist folglich nicht als Quantensprung anzusehen, sondern ein kontinuierlicher, partizipativer und iterativer, kritischer Prozess.

Benutzerbeteiligung ("User Involvement") ist der wichtigste Erfolgsfaktor in der Softwareentwicklung [4, 9]. Strauss [10] beschreibt beispielsweise User Boards, die sich in Entwicklungsprojekten als höchst sinnvoll und praktikabel erwiesen haben. In diesen User Boards, die sich aus Mitarbeitern des Projektteams inklusive Kunden und Benutzern zusammensetzen, werden Dialogentwürfe konstruktiv durchgespielt und bearbeitet. Gearbeitet wird dabei mit Papiausdrucken, Overhead-Folien und Post-Its.

5. USABILITY IST CHEFSACHE

Wie man bereits vom Qualitäts- und Projektmanagement weiß, lassen sich unbeliebte und scheinbar erst mal nur Arbeit erzeugende Tätigkeiten nur dann erfolgreich im Projekt einsetzen, wenn sie auch vom Management konsequent gewollt und verfolgt werden. Aufgabe des Managements ist es, Usability zu würdigen und die Forderung nach Usability bei den Entwicklern durchzusetzen. Ein gemeinsames Verständnis zu Usability ermöglicht allen Beteiligten zu erkennen, welchen Nutzen Usability neben einer guten Funktionalität für das Produkt, die Anwender und das Unternehmen bringt.

"Wir wollen eine benutzungsfreundliche Software" lässt sich als Anspruch erst dann durchsetzen, wenn entsprechende Ressourcen im Projekt vorgesehen

werden, wenn beispielsweise Usability Tests nicht als Luxus und zu kostspielig abgetan werden und wenn der Projektleiter auch bei Zeit- und Budgetdruck hinter den ursprünglich geplanten Maßnahmen steht und diese regelmäßig auf der Agenda hat.

6. DIE ROLLE DES USABILITY PROFESSIONALS: SCREEN VISAGIST ODER USABILITY CONSULTANT?

Die dargestellten Aspekte legen nahe, dass ein Usability Professional mehr können und sein muss als ein Screen Visagist.

Der Usability Consultant ist im Requirements Engineering genauso zu Hause wie in der Softwareergonomie, er hat umfassende Kenntnisse von Qualitäts- und Projektmanagement, er kann programmieren sowie präsentieren und Schulungen durchführen. Er berät User gleichermaßen wie Entwickler und das Management. Hilfreich sind Kenntnisse von Softwarearchitekturen (z.B. Komponentenbasierte Technologien, Corba, XML, Webservices). Er (oder sie) hat weitreichende soziale Kompetenzen, um mit unterschiedlichen Charakteren und Projektbeteiligten konstruktiv agieren zu können. Hilfreich sind natürlich auch Fach- und Branchenkenntnisse. Kurz gesagt: es ist ein Berater gefragt, der zu den Fragen des Software-Entwicklungsprozesses Stellung beziehen und Lösungen anbieten kann.

Usability lässt sich dann am erfolgreichsten umsetzen, wenn z.B. die Bewertung einer Benutzungsoberfläche oder die Entwicklung eines Style Guides Teil von projektbegleitenden und -übergreifenden Usability-Maßnahmen sind.

7. REFERENZEN

- [1] Nigel Bevan, Itzhak Bogomolni: Incorporating user quality requirements in the software development process, 4th International Software Quality Week Europe (Qwe2000)
<http://www.soft.com/QualWeek/QWE2K/Papers.pdf/Bevan.pdf>
- [2] DIN EN ISO 13407, Benutzerorientierte Gestaltung interaktiver Systeme, 2000/11
- [3] DIN EN ISO 9241-10, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Grundsätze der Dialoggestaltung, 1996/07
- [4] Steve McConnell: From the Editor - The Best Influences on Software Engineering. IEEE Software 17(1): 2000, p. 10-17
- [5] Ralf Fahney, Anforderungsmanagement als eigenständige Disziplin, GC-upa 2003
- [6] Ronald Hartwig, Praktische Integration des Requirements-Engineering im Feld der multi-medialen interaktiven Lehrmedien, GC-upa 2003
- [7] Michael Hatscher, Jörg Beringer: Customer-centered "New Application" Design, GC-upa 2003
- [8] ISO/TR 18529, Ergonomics of human-system interaction - Human-centred lifecycle process descriptions, 2000/06
- [9] Standish Group. "CHAOS Chronicles or CHAOS: A Recipe For Success", 1999
<http://www.standishgroup.com/chaos/intro1.php>
http://www.standishgroup.com/sample_research/PDFpages/chaos1998.pdf
- [10] Friedrich Strauß: Requirements-Analyse und GUI Design, GC-upa 2003
- [11] Volere 2003, <http://www.volere.co.uk/>

Referentin



Dipl.-Inform. M.Sc. Astrid Beck hat in Berlin Informatik und in Los Angeles Computer Science studiert. Zuvor hat sie eine Ausbildung als Informatik-Assistentin absolviert. In Stuttgart war sie bei der Universität Stuttgart in Kooperation mit dem Fraunhofer-Institut für Arbeitswirtschaft und Organisation tätig, bis sie sich 1994 selbstständig machte.

Astrid Beck ist seitdem als Beraterin aktiv für Software-Qualitätsmanagement.

Ihre Beratungsschwerpunkte sind die Gestaltung von Benutzungsoberflächen, Konzeption von Webprojekten sowie Methoden und Verfahren für die Softwareentwicklung, wie z.B. Anforderungsermittlung und Test. Ihre Kunden sind Banken und große Unternehmen.

ANFORDERUNGSMANAGEMENT ALS EIGENSTÄNDIGE DISZIPLIN

Ralf Fahney

Anforderungsmanagement

Helfrichstrasse 8

D-82041 Oberhaching

fahney@anforderungsmanagement.de

www.anforderungsmanagement.de

ABSTRACT

Studien belegen, dass professioneller Umgang mit Anforderungen ein Drittel bis die Hälfte des Erfolgs von erfolgreichen IT-Projekten ausmacht. Umgekehrt liegt es etwa zur Hälfte an mangelhaftem Anforderungsmanagement, wenn ein IT-Projekt schief läuft.

Anhand eines Praxisbeispiels und der gängigen Literatur motiviert dieser Beitrag, Anforderungsmanagement als eigenständige Disziplin in IT-Projekten zu sehen. Er zeigt das Entwicklungspotenzial für Unternehmen auf, die Anforderungsmanagement als Disziplin im Unternehmen etablieren.

Keywords

Anforderungsmanagement, Anforderungsanalyse, Requirements Management, Requirements Engineering, Change Request Management

1. EINLEITUNG

Besonders Großunternehmen sind von Einflüssen betroffen wie

- allgemeine wirtschaftliche Entwicklung;
- internationale Akquisitionen;
- gesetzliche und politische Vorgaben;
- wechselnden Verantwortlichkeiten im Unternehmen.

Es ist hier zum einen wichtig, den Umfang von IT-Projekten so zu vereinbaren, dass in kurzer Zeit Ergebnisse entstehen, die Nutzen erzeugen und weiter verwendbar sind. Zum anderen muss man Projekte gegen unkontrollierte und ungewollte

Veränderung der Anforderungen schützen.

Dieser Beitrag

- zeigt anhand eines Praxisbeispiels auf, wie klar vereinbarte Anforderungen stabile Rahmenbedingungen schaffen;
- motiviert anhand des Praxisbeispiels, Anforderungsmanagement als eigenständige Disziplin in IT-Projekten zu sehen;
- unterstützt diese Sichtweise durch Verweise und Zitate der Anforderungsmanagement-Literatur;
- zeigt das Entwicklungspotenzial für Unternehmen auf, die Anforderungsmanagement als Disziplin im Unternehmen etablieren.

2. PRAXISBEISPIEL

Aufgrund von Akquisitionen und gewachsenen Strukturen kommen im Kundenservice / Callcenter eines Logistik-Dienstleisters mehrere Auskunftssysteme zum Einsatz.

2.1 Projektziele

In einem Projekt will der Logistik-Dienstleister die Beauskunftung harmonisieren und dem Callcenter eine einheitliche Sicht auf Sendungen verschiedener datenliefernder Systeme zur Verfügung stellen (Harmonisierungsprojekt).

In einem zweiten Projekt will er das Reklamations- und Schadensmanagement stärker als bisher in den Arbeitsablauf im Callcenter integrieren.

Parallel zu diesen Aktivitäten entwickelt ein drittes Projekt die IT-Unterstützung für ein neues Produkt (Produkt-Neuentwicklung). Das Projekt formulierte einen Change Request an das Harmonisierungsprojekt mit dem Ziel, die Auskunftsfähigkeit im Callcenter um das neue Produkt zeitnah zu erweitern und die Pilotierung des neuen Produktes unterstützen zu können.

2.2 Pflichtenhefte

Der Logistik-Dienstleister lässt für die drei Aufgaben Pflichtenhefte erarbeiten. Hier sind wie üblich Masken, Dialogabläufe, Geschäftslogik, Schnittstellen usw. spezifiziert.

Es ist erlaubt digitale und Kopien in Papierform des ganzen Papers oder Teilen davon für den persönlichen Gebrauch oder zur Verwendung in Lehrveranstaltungen zu erstellen. Der Verkauf oder gewerbliche Vertrieb ist untersagt. Rückfragen sind zu stellen an den Vorstand des GC-UPA e.V. (Postfach 80 06 46, 70506 Stuttgart).

Proceedings of the
1st annual GC-UPA Track
Stuttgart, September 2003

© 2003 German Chapter of the UPA e.V.

In die regelmäßigen Abstimmungsgespräche sind der fachliche Auftraggeber, die Endbenutzer und die notwendigen Fachexperten eingebunden. Sie sind die wichtigsten Informationsgeber für die Spezifikation.

2.3 Reviewprozess

Gemeinsam mit dem Auftraggeber begutachtet die Projektleitung die Pflichtenhefte. Es ergibt sich, dass die Anforderungen in zwei Bereichen ausreichend stabil für eine Implementierung sind:

- Masken und Dialogabläufe für die Beauskunftung und für das Reklamations- und Schadensmanagement;
- Geschäftslogik zur Vereinheitlichung der Sendungsinformation.

Bei der Geschäftslogik für das Reklamations- und Schadensmanagement ergibt sich, dass die zentralen Vorgaben für die Gesamtarchitektur noch zu großen Spielraum für die Umsetzung lassen.

Für die Anbindung der datenliefernden Systeme ist zum einen das technische Design weiter zu konkretisieren. Zum anderen sind die Releaseplanungen der datenliefernden Systeme mit der Terminplanung des Harmonisierungsprojektes zu synchronisieren.

2.4 Beauftragung von Teillösungen

Die Auftraggeber wünschen noch stärker sichtbaren Projektfortschritt. Gleichzeitig ist Raum erforderlich für die Klärungen z.B. im Bereich der Gesamtarchitektur und der Schnittstellen zu den datenliefernden Systemen. Es entsteht die Idee, umsetzbare Teillösungen zu ermitteln und zu implementieren, die für sich genommen bereits Sinn im Hinblick auf die Projektziele ergeben.

In einem Klärungsprozess erarbeiten und vereinbaren alle Beteiligten zwei Teilprojekte, mit denen zum einen die gesamte Dialogoberfläche und zum anderen die Geschäftslogik zur Vereinheitlichung von Sendungen implementiert wird. Die Anforderungen aus dem Change Request des dritten Projektes sind so gestaltet, dass sie im Rahmen der beiden Teilprojekte mit umgesetzt werden können.

Der Logistik-Dienstleister beauftragt ein Softwarehaus mit der Durchführung.

2.5 Weitere Durchführung der Projekte

Der Umfang der beiden Teilprojekte und die umzusetzenden Anforderungen waren im Vorfeld so klar herausgearbeitet, dass der Dienstleister die Ziele innerhalb des geschätzten Zeit- und Budgetrahmens erreicht.

Parallel zur Durchführung der Teilprojekte werden die erforderlichen Klärungen für die Stabilisierung der verbleibenden Anforderungen fortgeführt. Aus den finalen Schnittstellenabstimmungen mit den

Datenlieferanten ergeben sich geringfügige Änderungen im Modell der vereinheitlichten Sendung und der bereits implementierten Dialogoberfläche. Die Pflichtenhefte werden entsprechend aktualisiert.

In Folgebeauftragungen erfolgte die planmäßige Umsetzung der vollständigen Lösung.

3. UNTERSUCHUNG DES PRAXISBEISPIELS

Dieser Abschnitt untersucht nun die Tätigkeiten der Projektleitung unter den Aspekten Projekt-, Qualitäts- und Anforderungsmanagement. Abbildung 1 setzt die drei Disziplinen schematisch zueinander in Beziehung. Sie verdeutlicht, dass es Überschneidungen und Abhängigkeiten gibt. Die Pfeile kennzeichnen den Informationsfluss zwischen den Disziplinen. Die Zahlen bezeichnen die Aspekte, die in den folgenden Unterabschnitten beleuchtet werden.

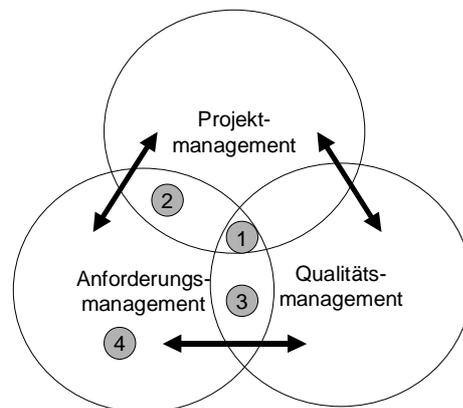


Abbildung 1: Verhältnis der Disziplinen Projekt-, Qualitäts- und Anforderungsmanagement zueinander

3.1 Gemeinsame Wurzel aller drei Disziplinen

Alle drei Disziplinen beginnen ihre Arbeit mit der Frage nach Projektauftrag und Projektziel. Anforderungsmanagement erhält initiale Orientierung für die weitere Klärung von Anforderungen.

3.2 Überschneidungen zwischen Anforderungs- und Projektmanagement

Anforderungsmanagement bedient sich der Methoden des Projektmanagements, um sein eigenes Vorgehen zu planen und zu koordinieren. Projektmanagement benötigt geklärte Anforderungen, um verlässliche Aufwands- und Zeitschätzungen erstellen zu können.

3.3 Überschneidungen zwischen Anforderungs- und Qualitätsmanagement

Zu jeder Anforderung sind z.B. die Abnahmekriterien festzulegen. Die Anforderungen selbst müssen qualitativ hochwertig formuliert sein (z.B. vollständig, widerspruchsfrei, verbindlich).