

UX meets RE

Hohe User Experience durch bedarfsgerechte Anforderungsspezifikation

Anne Gross

Fraunhofer Institut für Experimentelles Software Engineering (IESE)
Fraunhofer Platz 1
67663 Kaiserslautern
Anne.Gross@iese.fraunhofer.de

Steffen Hess

Fraunhofer Institut für Experimentelles Software Engineering (IESE)
Fraunhofer Platz 1
67663 Kaiserslautern
Steffen.Hess@iese.fraunhofer.de

Abstract

Die Erreichbarkeit einer hohen User Experience in Softwareprodukten setzt ein detailliertes Verständnis über Benutzer und deren Aufgaben voraus. Daher bilden Anforderungsdokumente für die Aktivitäten von Usability Professionals eine wichtige Arbeitsgrundlage. Aber auch für andere Rollen wie beispielsweise Architekten oder Tester bilden Anforderungsdokumente eine wichtige Informationsquelle. Dies bringt jedoch die Herausforderung mit sich, dass unterschiedliche Informationsbedarfe in diesen Dokumenten in Abhängigkeit der jeweiligen Rollen und Aufgaben abgedeckt sein müssen. Dabei ist zu beobachten, dass diese unterschiedlichen Informationsbedarfe häufig nicht adäquat adressiert werden, wodurch die Nutzbarkeit und auch Akzeptanz von Anforderungsdokumenten negativ beeinflusst wird.

In diesem Tutorial erarbeiten und diskutieren die Teilnehmer konkrete Informationsbedarfe und Erwartungen, die in Anforderungsdokumenten aus der Sicht von Usability Professionals adressiert sein sollten, um eine effektive und effiziente Nutzbarkeit der Dokumente zu erreichen. Basierend auf einem Framework zur systematischen Erhebung und Verfeinerung von Anforderungen werden resultierende Artefakte hinsichtlich ihrer Relevanz bewertet und geeignete Repräsentationen diskutiert.

Keywords:

/// User Experience
/// Usability Engineering
/// Requirements Engineering
/// Anforderungsdokumente
/// Perspektiven-orientierte Anforderungsspezifikation

1. Motivation

Die Entwicklung von Software mit einer hohen User Experience (UX) geht nicht ohne eine detaillierte Erfassung und Analyse von Benutzeranforderungen einher. Daher bilden Dokumente, die diese Anforderungen dokumentieren, eine wichtige Grundlage für die Aktivitäten eines Usability Professionals: etwa zur benutzer- und aufgabenorientierten Gestaltung von Bedienoberflächen oder von Interaktionen.

Aber nicht nur für Usability Professionals gelten Anforderungsdokumente als eine wichtige Informationsquelle. Auch andere Rollen, welche in Softwareentwicklungsprojekten involviert sind, wie Architekten, Tester und Programmierer, basieren ihre Arbeiten und Aktivitäten auf Informationen und Artefakten, die in Anforderungsdokumenten spezifiziert sind.

Dies führt zu der Herausforderung, dass Anforderungsdokumente unterschiedliche Informationsbedürfnisse und Erwartungen adressieren müssen, die abhängig sind von der jeweiligen Rolle und Aufgabe, die die Leser der Anforderungsspezifikationen (sog. „Dokument-Stakeholder“) haben. Abbildung 1 illustriert verschiedene Dokument-Stakeholder und veranschaulicht exemplarische Informationsbedarfe für Tester, UI Designer und Architekt. Ein Architekt benötigt z. B. detailliertes Wissen über Qualitätsanforderungen, Daten und technische Rahmenbedingungen, wohingegen die Arbeiten eines Usability Professionals auf detaillierten Beschreibungen der Benutzer, Use Cases, sowie zu unterstützenden Aufgaben und Workflows basieren. Für den Tester stellen ebenfalls detaillierte Use Case Beschreibungen eine wichtige Informationsgrundlage dar, etwa zur Ableitung von Testfällen.

Aber nicht nur bezüglich der Relevanz verschiedener „Anforderungstypen“ (Artefakten) wie z. B. Benutzerbeschreibungen oder Workflow-Beschreibungen gibt es unterschiedliche Bedarfe zwischen den Dokument-Stakeholdern. Ebenso gibt es Unterschiede hinsichtlich der geeigneten Repräsentation der verschiedenen Artefakte (z. B. Notation, Detailtiefe) was eine initiale Studie gezeigt hat (siehe Abschnitt 2.2).

[Abb. 1]

Heutzutage gibt es bereits eine Vielzahl etablierter Methoden, Standards, und Guidelines, die Anforderungsingenieure dabei unterstützen, Anforderungen systematisch zu erheben und zu spezifizieren [1, 2, 3, 4]. Allerdings adressiert keine dieser Methoden gezielt und explizit die „individuellen“, d.h. rollen-spezifischen Informationsbedarfe der unterschiedlichen Dokument-Stakeholder. Vielmehr resultieren daraus Anforderungsdokumente, die weit mehr Informationen enthalten, als der

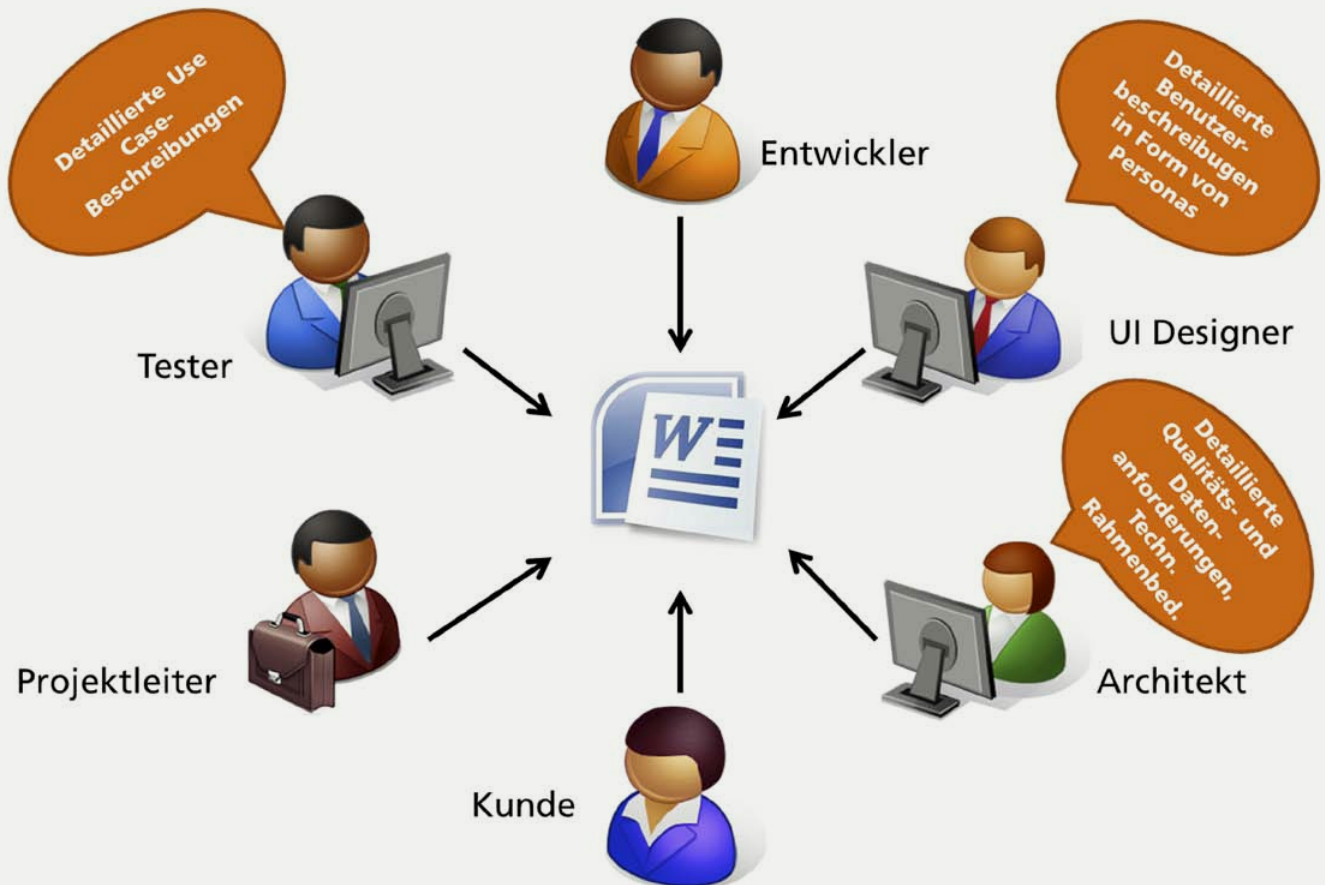


Abb. 1.
Dokument-Stakeholder und
ihre Informationsbedarfe

Dokument-Stakeholder für die Durchführung seiner Aktivitäten eigentlich benötigt, was das Auffinden relevanter Informationen im Dokument erschwert. Oder aber es fehlen wichtige Informationen, oder die Informationen sind unzureichend repräsentiert etc. [5].

Alle diese Faktoren beeinflussen auf negative Weise eine effiziente und effektive Nutzung der Anforderungsdokumente, was zur Folge hat, dass die Akzeptanz für solche Anforderungsdokumente negativ beeinflusst wird. Im schlimmsten Fall führt diese geringe Akzeptanz dazu, dass Anforderungsdokumente von den Dokument-Stakeholdern vernachlässigt

werden und es letztendlich zur Umsetzung von Softwareprodukten kommt, die die Benutzeranforderungen nicht adressieren.

Mit zunehmender Komplexität von Softwaresystemen und Trends hinsichtlich agiler Entwicklung wird dieses Problem der unzureichenden Nutzbarkeit und Akzeptanz von Anforderungsdokumenten in Zukunft ein immer größer werdendes Problem darstellen. Im Folgenden werden zunächst die Lösungsidee von perspektiven-orientierten Anforderungsspezifikationen sowie bisherige Aktivitäten vorgestellt. Anschließend wird noch ein kurzer Ausblick auf aktuelle und künftige Aktivitäten gegeben.

2. Perspektiven-orientierte Anforderungsspezifikationen

Um dem oben beschriebenen Problem der unzureichenden Nutzbarkeit von Anforderungsdokumenten entgegenzuwirken, erforschen die Autoren derzeit „Perspektiven-orientierte Anforderungsspezifikationen“ mit dem Ziel, den Dokument-Stakeholdern bedarfsgerechte Anforderungsdokumente zur Verfügung zu stellen, die effektiv und effizient genutzt werden können (vgl. [Abb. 2]).



Abb. 2.
Forschungsvorhaben zur
perspektiven-orientierten
Anforderungsspezifikation

In einem ersten Schritt arbeiten die Autoren derzeit daran, empirisch validiertes Wissen darüber zu erhalten, welche Informationsbedarfe aus der Perspektive der verschiedenen Dokument-Stakeholder in Abhängigkeit ihrer jeweiligen Rolle und Aufgaben im Rahmen eines Softwareentwicklungsprojektes existieren. Im Folgenden wird dabei besonders auf die Rolle des Usability Professionals eingegangen und somit folgende Forschungsfragen adressiert:

- Welche Aufgaben eines Usability Professionals werden mit Hilfe von Anforderungsdokumenten unterstützt?
- Welche Artefakte sind für die Durchführung dieser Aufgaben relevant?
- Wie sollen diese Artefakte repräsentiert sein?
 - Welche Notation wäre optimal für die Darstellung?
 - Welche Details / Informationen sollen dokumentiert werden?

In einem zweiten Schritt soll das Wissen über die jeweiligen Informationsbedarfe

in ein Wissensmodell einfließen, welches es schließlich durch geeigneten Toolsupport ermöglicht, perspektiven-orientierte Anforderungsspezifikationen zu generieren, beispielsweise durch Bildung von perspektiven-orientierten „Sichten“ auf ein Anforderungsdokument. Dies ermöglicht es, dass den Dokument-Stakeholdern alle (und nur) relevante Artefakte und Informationen im Anforderungsdokument zur Verfügung gestellt werden, die diese benötigen, um ihre Aufgaben optimal zu erfüllen.

Eine solche Lösung bildet somit auch aus Sicht von Usability Professionals einen wesentlichen Vorteil, da für die Entwicklung und Gestaltung von Software mit einer hohen UX eine detaillierte Analyse der Anforderungen seitens des Usability Professionals eine unabdingbare Voraussetzung ist. Eine bedarfsgerechte und perspektiven-orientierte Anforderungsspezifikation kann hier eine wesentliche Unterstützung bieten, Softwareprodukte mit einer höheren UX effektiver und effizienter entwickeln können.

3. Bisherige Arbeiten

3.1. TORE Framework

Die derzeitigen Forschungsarbeiten zur Erarbeitung der verschiedenen Informationsbedarfe basieren auf dem am Fraunhofer IESE entwickelten TORE Framework [6]. Dieses Framework (dargestellt in Abbildung 3) umfasst verschiedene Entscheidungspunkte, die einen Anforderungsingenieur darin unterstützen, Anforderungen systematisch zu erheben und über verschiedene Abstraktionsebenen hinweg zu verfeinern. Ausgehend von der Entscheidung, welche Benutzer vom Softwareprodukt unterstützt werden sollen, spezifiziert man deren Ziele und Aufgaben und verfeinert diese Aufgaben systematisch weiter über eine Analyse von Workflows bis hin zu konkreten Systemfunktionalitäten und Interaktionsbeschreibungen, die von verschiedenen Stakeholdern weiterverarbeitet werden. [Abb. 3]



Abb. 3.
Entscheidungspunkte des
TORE Frameworks



Stefanie

„Jetzt werde ich allein wohnen. Ich muss alle meiner Ausgabe sehr gut kontrollieren, da ich noch Studenten bin und nicht viel Geld zur Verfügung habe.“

Stefanie ist Student von Biologie an der Uni-KL, bis jetzt hat sie mit ihrer Schwester gewonnen. In die nächste Woche zieht ihre Schwester aus. Sie muss die Wohnung jetzt allein verwalten. Für Stefanie ist wichtig das Thema Ökologie, aber sie hat auch nicht viel Geld, deswegen beiden Sachen müssen neben einander sein.

Alter: 21
Beruf: Student von Biologie:
Familienstatus: Ledig, kein Kind.
Einkommen: 12.200 €/Jahr (Stipendium + Nebenjob)
Technische Profile: Sie kennt sich ziemlich gut mit Computer und hat ein MacBook Pro mit OS X + iLifeseit 6 Monate;
Wireless-Internetverbindung: 20-25 Stunde/Woche online.
Internetnutzung: 100% zu Hause, Unterhaltung, Shopping, Chat, News und Information.
Liebungsseiten:

Abb. 4.
Persona (Beispiel)

Table 44: UC_DP3: Forward information to rescue vehicle

Use Case Number	UC_DP3
Use Case Name	Forward information to rescue vehicle
Actors	Dispatcher
Version Number	V1.0 (F. Kaiser, E. Otto, A. Gross)
Author	Final
Iteration	
Summary	The Dispatcher contacts the emergency doctor / paramedic and forwards relevant information.
Trigger/ intent	Dispatcher found a suitable rescue vehicle to handle the emergency and now wants to give them proper information.
Supported goal(s) from WP2	- Keep 15-minute max. time limit for assistance (DP_G1) - Keep 90 sec. time limit until alert of response vehicle (DP_G2) - Forward information about proper access to apartment (DP_G3) - Provide important data, medical information to emergency doctor / paramedic (DP_G12)
Supported workflow	Workflow dispatcher: Forward alarm (part 2) (see Figure 52)
Supported task(s) from WP2	TOBE_DP_12: Forward information to rescue vehicle
Preconditions	- System Dispatcher found suitable rescue vehicle - Device in vehicle (Edoc-subsystem) is in operation
Flow of events:	
Main flow	- Dispatcher contacts suitable rescue vehicle. - System establishes connection between dispatcher and rescue vehicle. - Dispatcher forwards the necessary information to the rescue vehicle. - System transmits the information and gives feedback
Alternative flows	N/A
Exceptional flows	N/A
Assumptions/ rules	Data transmission is successful
Displayed information from User Model (see also [BS+08])	N/A
Postconditions	All relevant information has been forwarded to the rescue vehicle.
NFRs	Efficiency Response Time: The system should establish a connection to the rescue vehicle within 30 seconds. Transmission Time: Alarm and data should be forwarded within 30 seconds. Usability Appropriate feedback: The system should inform about current states of activities that are executed, e.g., in case that data has been sent (see SF_UC_DP3_3)
System Functions	SF_UC_DP3_1: Establish connection to rescue vehicle SF_UC_DP3_2: Transmit information SF_UC_DP3_3: Inform that data has been sent
Relation to other use cases	N/A
Open issues	N/A

Abb. 5
Beispiel für eine tabellarische Use Case Beschreibung auf Basis eines Templates

Wendet man dieses Framework nun in einem konkreten Softwareprojekt an, so entstehen verschiedene Artefakte (wie z. B. Benutzerbeschreibungen, Zielbeschreibungen, Workflow-Beschreibungen, Interaktionsbeschreibungen), die die getroffenen Entscheidungen in Form von konkreten Anforderungen dokumentieren. Zur Dokumentation dieser Artefakte stehen verschiedene Notationen zur Verfügung. Abbildung 4 illustriert beispielsweise eine Benutzerbeschreibung in Form einer Persona [7], Abbildung 5 eine Interaktionsbeschreibung in Form einer tabellarischen Use Case Beschreibung [8].

[Abb. 4] [Abb. 5]

3.2. Eye-Tracker Studie

3.2.1. Ablauf

In einer initialen Eye-Tracker Studie mit einem mobilen Eye-Tracker (vgl. Abbildung 6 und Abbildung 7) wurden 2 Software Architekten und 2 Usability Professionals mit einer Anforderungsspezifikation bestehend aus einer Spezifikation der Systemanforderungen und einer Spezifikation von Domänenanforderungen konfrontiert. Die Dokumente wurden basierend auf dem TORE Framework erstellt und enthielten insgesamt 35 Artefakte, die aus Sicht des Anforderungsingenieurs für die Erstellung des Systems relevant sind. Das Verhalten der Probanden wurde durch Eye-Tracking, eine externe Kamera und 1-3 Beobachter, die jeweils Protokoll führten analysiert.

Außerdem war ein Versuchsleiter anwesend und die Probanden waren angewiesen laut zu denken. Ziel der Studie war es, grundsätzliche Informationsbedarfe der beiden Rollen zu erheben und darüber hinaus auch Unterschiede in der Arbeitsweise mit dem Dokument festzustellen. Die Abbildungen der Prozesse waren auf Grund der Größe teilweise im Raum aufgehängt, weswegen ein mobiler Eye-Tracker die optimale technische Unterstützung zur Analyse war.

Basierend auf der Auswertung der Studie wurden Schlussfolgerungen über Schwachstellen im Dokument und Relevanz der Artefakte gezogen.



Abb. 6.
Eye-Tracker Studie - Bearbeitung
des Anforderungsdokumentes



Abb. 7.
Eye-Tracker Studie -
Workflowabbildungen

3.2.2. Ergebnisse

Im Rahmen der Eye-Tracker Studie wurden die Teilnehmer zusätzlich in Form eines Fragebogens zu Problemen, Relevanz und Verwendungszweck aller Artefakte befragt. Dabei wurde vor allem festgestellt, dass nicht alle benötigten Artefakte in den Anforderungsdokumenten enthalten waren (z. B. technische Rahmenbedingungen fehlten). Es wurde ebenso deutlich, dass nicht alle Artefakte für die Rollen Architekt und Usability Professional gleichermaßen relevant sind und dass es starke Unterschiede in der Präferenz zwischen den einzelnen Rollen gibt. Tabelle 1 zeigt die Auswertung der Artefakt-Relevanz in Bezug zur jeweiligen Rolle und zeigt auf den ersten Blick, dass für den Usability Professional eine erhebliche Menge an Artefakten

keine Relevanz hat und dadurch die Arbeit mit dem Dokument unnötig erschwert.

[Tab. 1]

Darüber hinaus wird schnell deutlich, dass ein Anforderungsdokument einen Kompromiss darstellt, mit dem Architekten und Usability Professionals zwar arbeiten können, aber für keine der beiden Rollen ist das Dokument wirklich optimal. Im Folgenden werden diese unterschiedlichen Informationsbedarfe anhand von konkreten Beobachtungen während der Studie für ausgewählte Artefakte erläutert.

Allgemeines

Aus Sicht der Usability Professionals beinhaltet ein Anforderungsdokument zu viele Informationen. Es werden nur diese Informationen gelesen, die als tatsächlich relevant erachtet werden. Problem dabei ist es, diese Informationen im Dokument zu finden, wenn es nicht angemessen strukturiert ist. Klassische Vorgehensweise ist, dass zunächst auf Personas, Use Cases und Informationen zum Nutzungskontext zugegriffen wird. Für Architekten beinhaltet das Anforderungsdokument zu viel textuelle Beschreibung, eine Darstellung primär in Form von Diagrammen wäre sinnvoll. Wichtig ist eine gute Strukturierung und Formatierung des Dokumentes, da der Architekt zunächst versucht, sich einen Überblick über das zu erstellende System zu verschaffen, in dem er vor allem auf Überschriften, Grafiken und Diagramme zugreift – wichtige Dinge markiert und diese später im Detail anschaut.

Stakeholder Beschreibungen

Usability Professionals sind sehr stark an die strukturierte Persona gebunden. Diese hilft ihnen deutlich mehr als eine rein textuelle Beschreibung der Benutzer.

Architekten hingegen interessieren sich überhaupt nicht für Personas, da Details der Stakeholder keine Relevanz besitzen. Wichtige Attribute der Stakeholder bezüglich der Systemanforderungen (z. B. Erwartungen ans System, Involvierung ins System) sollten vielmehr in Form von Tabellen oder Listen dargestellt werden, um einen schnellen Überblick zu gewinnen.

Use Cases

Eine gute Verfolgbarkeit zwischen Use Case Diagrammen und Use Case Beschreibungen ist für Usability Professionals eine wichtige Eigenschaft, um mit dem Anforderungsdokument produktiv arbeiten zu können. Im Use Case ist vor allem die Beschreibung des Interaktionsflusses relevant. Nichtfunktionale Anforderungen (insbesondere solche, welche sich auf Usability beziehen) beschreiben in der Regel Sachverhalte, die man als guter Usability Professional sowieso berücksichtigen sollte und sind insofern nicht hilfreich, da sie nicht detailliert spezifiziert sind. In der durchgeführten Studie waren Architekten der Auffassung, dass Use Case Diagramme nicht so hilfreich wie Workflows sind. In einem Use Case sind außerdem zu viele detaillierte Informationen enthalten, die es schwierig machen einen Überblick zu gewinnen. Um als Architekt optimal damit arbeiten zu können, sollten Use Case Beschreibungen auf die wesentlichen Elemente (Interaktionsfluss, Nichtfunktionale Anforderungen und Daten) reduziert werden.

Workflows

Bei der Darstellung der Workflows waren sich Usability Professionals und Architekten einig, dass eine grafische Darstellung der Workflows sehr hilfreich ist, dass diese aber unbedingt auch durch eine textuelle Beschreibung ergänzt werden sollte, um

Relevanz der Artefakte	Architekt	Usability Professional
Sehr wichtig	19/35	18/35
Eher wichtig	12/35	4/35
Eher unwichtig	4/35	4/35
Unwichtig	0/35	9/35

Tab. 1.
Relevanz der Artefakte

ggf. Missverständnissen vorzubeugen und unklare Sachverhalte nachlesen zu können. Hierbei sollte dann auf unnötige Details, wie z. B. Probleme und Wünsche direkt im Workflowdiagramm verzichtet werden und dort eine möglichst übersichtliche Darstellung geboten werden.

Nichtfunktionale Anforderungen (Qualitätsanforderungen)

Zur Erstellung des User Interface Designs sind nicht alle im Anforderungsdokument spezifizieren Qualitäten relevant. Positiv hervorgehoben wurde, dass die Anforderungen sehr strukturiert dargestellt sind und man so schnell die für UI Design relevanten Qualitäten findet.

Für die Architekten hingegen sind alle dargestellten Qualitäten relevant. Vor allem die graphische Repräsentation und die Verfeinerung der Attribute bieten hier dem Leser einen Mehrwert. Die zusätzliche textuelle Beschreibung erweckt den Eindruck, dass hier redundante Informationen gezeigt werden. Wichtige Informationen sollten auf jeden Fall im Text hervorgehoben sein.

Darüber hinaus wurden in der Studie weitere allgemeine Erkenntnisse gewonnen:

- Die Relevanz der Artefakte ist von der Projektsituation abhängig, d.h. eine flexible Lösung wird benötigt
- Die Vorgehensweise bei der Analyse der Dokumente hängt sehr stark von der individuellen Person ab
- Systemanforderungen sind generell relevanter als Domänenanforderungen
- Verbesserung bezüglich der Notationen sollte in Betracht gezogen werden

4. Future Work

Künftige Arbeiten haben in erster Linie das Ziel, die Ergebnisse der initialen Studie zu validieren und somit eine größere Datengrundlage für die Entwicklung der perspektiven-orientierten Anforderungsspezifikation zu erhalten. Hierzu wird als nächste Aktivität ein Tutorial auf der UPA 2011 durchgeführt. Weiterhin soll die Erstellung von perspektiven-orientierten Anforderungsspezifikationen durch ein Tool unterstützt werden, um somit gleichermaßen dem Anforderungsingenieur als auch den weiteren beteiligten Rollen (wie Architekt und Usability Professional) eine wertvolle Unterstützung bei der Entwicklung von Softwaresystemen zu bieten. Anforderungsingenieure würden so bei der Erhebung der Informationen und der Erstellung der Anforderungsdokumente unterstützt werden. Architekten und Usability Professionals würden eine Arbeitsgrundlage erhalten, die jeweils auf ihre Bedarfe zugeschnitten ist und in produktivem Einsatz optimal verwendet werden könnte.

Literatur

1. IEEE (1998). Recommended practice for Software Requirements Specifications. Standard IEEE Std. 830-1998, Institute of Electrical and Electronic Engineers, 1998
2. Volere Requirements Specification Template: Robertson S., Robertson, J (2006). Mastering the Requirements Process—Second Edition, Addison-Wesley, 2006. ISBN 0-321-41949-9
3. Simon, T., Streit, J., and Pizka, M. (2008). Practically Relevant Quality Criteria for Requirements Document, itestra GmbH, Germany, published at International Conference on Software Engineering Research and Practice 2008
4. Firesmith, D. (2003) "Modern Requirements Specification", in Journal of Object Technology, Vol. 2, No. 1, March-April 2003, pp. 53-64
5. Gross, A. (2010). Perspective-based Specification of Efficiently and Effectively Usable Requirements Documents", Doctoral Symposium, Requirements Engineering Conference (RE'10), Sydney, Australia
6. Adam S., Doerr J., Eisenbarth M., and Gross A. (2009). Using Task-oriented Requirements Engineering in Different Domains - Experiences with Application in Research and Industry. In Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE (RE ,09). IEEE Computer Society, Washington, DC, USA, 267-272, 2009
7. Cooper, Alan. The Inmates Are Running the Asylum : Why High-Tech Products Drive Us Crazy and How to Restore the Sanity. Indianapolis, USA, 1999.
8. Jacobson I. (2004). Use cases -- Yesterday, today, and tomorrow" in Software and systems modeling, pp 210-220, August 2004