
A.1. SoftWiki - Agiles Requirements-Engineering für Softwareprojekte mit einer großen Anzahl verteilter Stakeholder

Sören Auer^{1,2}, Thomas Riechert¹, Klaus-Peter Fähnrich¹

¹ *Universität Leipzig, Institut für Informatik, Abteilung Betriebliche Informationssysteme*

² *University of Pennsylvania, Computer and Information Science Department*

1. Einleitung

In den 80er und 90er Jahren hatten große Anwendungssysteme in Unternehmen einige hundert bis tausend Anwender. Der Software-Entwicklungsprozess für diese Anwendungen war innerhalb der Unternehmen relativ klar geregelt. Fachinformatiker und Fachabteilungen standen einander dabei gegenüber. Oft wurden auch externe Fachleute und Komponentenlieferanten integriert. Entwicklungsmethoden und Werkzeuge waren auf diese Situation ausgelegt. Seit dieser Zeit haben wesentliche Veränderungen stattgefunden. Internettechnologien haben neue Klassen von Applikationen ermöglicht, die wie folgt charakterisiert werden können:

- Die Applikationen sind kooperativ (unternehmensübergreifend). Nicht selten sind 20-50 oder mehr Unternehmen z.B. bei Zulieferketten beteiligt.
- Eine eigene Klasse bilden mandantenfähige Systeme sowie Business-to-Consumer Systeme (B2C) bei denen sehr große Nutzerzahlen konnektiert werden.
- Die Entwicklungszeiten liegen im Bereich von Monaten statt Jahren für eine erste Bereitstellung einer Basislösung.
- Die Systeme werden inkrementell unter starker Anwenderbeteiligung bis hin zur Endbenutzerbeteiligung weiterentwickelt.

Diese Systeme werden oft unter Verwendung von Applikationsplattformen entwickelt, die selber in einem parallel laufenden Prozess weiterentwickelt werden. Beispiele für solche Applikationen sind E-Business Systeme, E-Government Systeme, E-Learning Systeme, Supply-Chain-Management-Systeme, Telekom-Mehrwertdienste und Telematikdienste.

Bei ihrer Entwicklung treten folgende Problemstellungen auf:

- Das Requirements-Engineering ist komplex, da viele unterschiedliche Stakeholder berücksichtigt werden müssen.
- Spezifikation und Entwicklung erfolgen oft weit verteilt; für Systemintegrationen sind viele unterschiedliche Partner notwendig.
- Endbenutzer in großer Zahl sollten in Feedback-Schleifen eingebunden werden.

- Es sind keine adäquate Methoden für das parallele Engineering von Plattformen und darauf aufbauenden Applikationen bekannt.

Das Requirements-Engineering für diese Applikationen wird insbesondere dadurch komplex, dass die Applikationen oft zentral für komplexe Dienstleistungs- und soziotechnische Systeme sind, wie dies der Fall von E-Government Diensten verdeutlicht. Der hier vorgestellte Ansatz versucht, das Thema als Kommunikation großer Gruppen zu begreifen und diese Kommunikation bei der Extraktion, Fortschreibung, Versionierung, Validierung, Moderation und Kategorisierung von Anforderungen als auch die Verbindung zu den nachgelagerten Prozessen Spezifikation, Codierung, Test und Projektmanagement durch wissensbasierte Schnittstellen zu unterstützen.

2. Ausgangssituation

In der Praxis eingesetzte Ansätze für das Requirements-Engineering sind stark textuell ausgerichtet und beruhen teils auf „flachen“ Dokumenttypen. Diese Dokumenttypen weisen nur rudimentäre Strukturierungen wie Kapitel und Absätze auf. Die Möglichkeiten der eingesetzten Werkzeuge werden häufig nicht genutzt, da Nutzern Funktionalitäten nicht in Bezug zum Requirements-Engineering dargebracht werden und dadurch existierende Methodik, die zugehörigen Prozesse und die Werkzeuge für viele Anwender zu komplex erscheinen.

Ein wichtiges Paradigma des modernen Requirements-Engineering ist die zielbasierte Vorgehensweise [YuMy98]. Hierbei werden ausgehend von den Absichten der relevanten Stakeholder, die Anforderungen abgeleitet, die zur Befriedigung der Ziele erfüllt werden müssen [La01]. Zur Dokumentation von Zielen wurden bis heute eine Reihe von Ansätzen vorgeschlagen, beispielsweise i-Star [Yu97] oder dessen „Nachfolger“ GRL [GRL05]. Die oben genannten Ansätze wurden für traditionell übliche Rahmenbedingungen des Requirements-Engineering erarbeitet, insbesondere eine moderate Zahl von Stakeholdern sowie langfristige Entwicklungszyklen. Die spezifischen Aspekte und Herausforderungen einer kooperativen Erhebung, Dokumentation und Validierung der Anforderungen einer sehr großen Zahl von Stakeholdern (größer 1000) wurde bisher nicht genauer untersucht.

Die verschiedenen Teilprozesse werden für gewöhnlich isoliert voneinander durchgeführt und dann ad-hoc zueinander in Beziehung gesetzt.

Auch in der Wissenschaft wurden diese Teilaspekte bisher größtenteils isoliert betrachtet. So wurden zwar erste Ansätze zur Integration unterschiedlicher Anforderungstechniken (also von Notationen und Konzepten) erarbeitet [Ol04], allerdings ohne die Berücksichtigung des Requirements-Management oder der

angrenzenden Prozesse. Umfassendere Ansätze zur Systematisierung und Integration dieser Prozesse wurden im Rahmen des ITEA-Projektes FAMILIES mit dem Fokus auf der Produktlinienentwicklung ausgearbeitet. Die grundlegenden Aspekte des Requirements-Management werden in dem dort erarbeiteten Informationsmodell berücksichtigt und mögliche Nachvollziehbarkeitsinformationen von Anforderungen definiert. Die Wechselwirkungen zwischen anderen Teilprozessen werden allerdings nur generisch betrachtet, ohne die konkreten Ausprägungen entsprechender Prozesse zu betrachten wie z. B. im Falle von Risiko- oder Änderungsmanagement.

Insbesondere für ein verteiltes und kollaboratives Requirements-Engineering ergeben sich also hier gänzlich neue Herausforderungen.

3. Basistechnologien für agiles Requirements-Engineering

Das SoftWiki-Konzept kombiniert Ansätze und Methoden aus drei verschiedenen Forschungsgebieten. Requirements-Engineering, in welchem Methoden zur Effektivierung und Präzisierung der Erfassung und Beschreibung von Anforderungen entwickelt werden. Um große weit verteilte Stakeholdergruppen zu befähigen in einem Requirements-Engineering-Prozess zusammenzuarbeiten kommen Konzepte des Semantic-Web zum Einsatz. Nicht zuletzt sind im Gebiet Knowledge-Management Theorien und unterstützende Softwaresysteme entwickelt worden, um Wissen zwischen verschiedenen Beteiligten strukturiert auszutauschen und weiterzuverarbeiten. Die SoftWiki-Idee ist darüber hinaus stark inspiriert von Kent Becks agiler Softwareentwicklungsmethode Extreme Programming [BeAn04] und Ward Cunninghams Community-Content-Management Ansatz der WikiWikiWebs [LeCu01].

3.1 Semantic-Web

Das Semantic-Web zielt auf eine Erweiterung des existierenden Webs, in welchem Informationen eine wohldefinierte Semantik zugeordnet wird mit dem Ziel Menschen und Computer zu befähigen besser zusammenzuarbeiten [Be01]. Es wird ein allgemeines Framework bereitgestellt, welches es erlaubt Daten über Applikations-, Unternehmens- und Communitygrenzen hinweg zu nutzen und wieder zu verwenden.

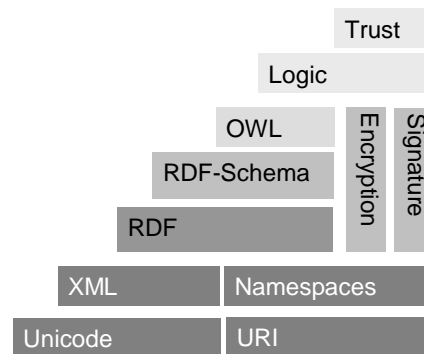


Abbildung 1: Semantic-Web Schichtenmodell [W3C01]

Erreicht werden soll dies durch Standards und Technologien, die im Rahmen der Semantic-Web-Initiative des WorldWideWeb Konsortiums (W3C) entwickelt wurden (siehe auch Semantic-Web Schichtenmodell in Abbildung 1).

Wichtige Semantic-Web Konzepte auf welche SoftWiki aufbaut sind:

- standardisierte Kodierung für Daten und Dokumente im Web (Unicode/UTF),
- weltweit eindeutige Identifizierung und Referenzierung von Entitäten durch Universal Resource Identifier (URI),
- einfaches Datenmodell welches an Subjekt-Prädikat-Objekt Aussagen natürlicher Sprache angelehnt ist (RDF),
- eine darauf aufbauende semantische Hierarchie von Vokabularen (RDF-Schema, OWL, domänenspezifische und Upper Level Ontologien).

Die Web Ontology Language (OWL - [Be04]) basiert darüber hinaus auf einer Beschreibungslogik, die es ermöglicht, Klassifizierungen, Subsumptionsberechnungen und Konsistenzchecks von und in OWL Wissensbasen automatisiert durchzuführen. Das SoftWiki-Konzept ist eng mit diesen Ansätzen und der Nutzung und Integration bestehender Entwicklungen, wie z. B. Description Logic Reasoner [Ho98], Upper Level Ontologien, verzahnt.

Techniken des Semantic-Web beginnen sich derzeit langsam durchzusetzen. Schwerpunkte sind insbesondere Bereiche in denen auf große Datenbestände weltweit verteilt zugegriffen und selbige integriert werden müssen (Semantic Grid), soziale Netzwerke (z.B. FOAF [BrMi05]) und verteiltes Content- und Wissensmanagement.

3.2 Knowledge-Management und Wiki-Systeme

In letzter Zeit werden konventionelle, schwergewichtige Knowledge-Engineering und -Management Systeme in zwei Richtungen ergänzt. Dies sind auf der einen Seite Ansatzpunkte zur Unterstützung früher, kreativer Phasen der Systementwicklung und andererseits um semantische Konzepte angereicherte Content Management und Community-Kollaborations-Technologien.

Ansatzpunkte zur Unterstützung früher, kreativer Phasen der Systementwicklung liegen in sehr unterschiedlichen Feldern wie Groupware-Systemen, neuen Interaktionstechniken zur Modellierung oder automatisierten Verfahren zur Analyse vorhandener Dokumentationen. Zur Unterstützung von Gruppensitzungen durch die Entwicklung gemeinsamer Themenstrukturen existieren bereits seit längerem Ansätze, bei denen das Wissen der Gruppe explizit formuliert und durch visuelle Sprachen modelliert wird. Entsprechende Systeme sind u.a. gIBIS [St94], Dolphin [St94] und DISCBOARD [Ga03]. Diese Systeme erfordern erheblichen zusätzlichen Erstellungsaufwand und haben noch keine breite praktische Anwendung gefunden. Ausnahmen sind einfache kommerzielle Brainstorming-Systeme wie z.B. MindManager. Ein Ansatz, Themen aus der Konversation in Gruppen zu extrahieren, wird von diMicco und Bender [DiBe04] verfolgt. Auch hier erfolgt eine Klassifikation von Themen anhand vordefinierter Kategorien. Insgesamt existieren einzelne, aus unterschiedlichen Forschungsbereichen stammende Ansätze, die das Potential bieten, die Unterstützung früher Phasen zu verbessern. Diese Ansätze sind allerdings für den breiteren Einsatz meist nicht ausreichend entwickelt und insbesondere bislang nicht integriert.

Wiki Technologien erfreuen sich zunehmender Beliebtheit zur kollaborativen Web gestützten Erstellung von Texten. Für das Problem mangelnder Strukturierung, Annotation und semantischer Formalisierung existieren mit [Kr05] und [Au05] erste Ansätze. Neben der Integration semantischer Auszeichnungen mittels Einbettung von RDF Statements direkt in den Wiki-Texten existieren auch Ansätze die Speicherung semantischer Informationen separat vorzunehmen [Ta04] und die Anlehnung an das Wiki-Systeme mehr auf konzeptueller Ebene zu verstehen.

4. Das SoftWiki-Konzept

Das SoftWiki-Konzept soll einen entscheidenden Beitrag zur Implementierung iterativer, agilerer terminologiebildender und erkenntnisgewinnorientierter Requirements-Engineering Prozesse leisten. Erreicht wird dies durch die Betonung und Unterstützung eines evolutionärerem Requirements-Engineering. Bestehende Wikisysteme sind hierfür nicht adäquat einsetzbar, da sie keinerlei semantische Strukturierungsmöglichkeiten bieten. Die feingradige terminologische Strukturierung von Requirements bietet gänzlich neue Anwendungsperspektiven von Feedback- und Abstimmungsmechanismen, sowie Peer- bzw. Community-Reviews für das Requirements-Engineering und damit eine wesentliche Steigerung der Anwenderorientierung bei der Softwareentwicklung.

Die SoftWiki-Lösungsarchitektur ist in Abbildung 2 visualisiert und Schwerpunkte werden im Folgenden detaillierter ausgeführt. Zentral für das Konzept ist das Softwiki-Werkzeug als Repository für die Anforderungen, Produktfunktionen und Fehler einer Software-Applikationen. Über Adapter ist die Ankopplung an CASE-Werkzeuge, welche Software-Entwicklungsprozesse implementieren, möglich. Umfangreiche Möglichkeiten der Visualisation ermöglichen die Interaktion verschiedener Stakeholder mit Werkzeug.

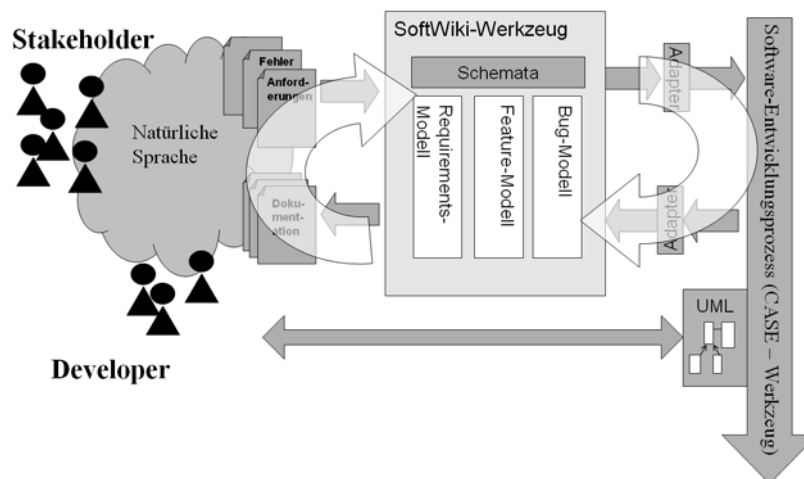


Abbildung 2: SoftWiki-Lösungsarchitektur

4.1 Semantisches Requirementsmodell

Ein zentrales Ziel ist die Unterstützung der semantischen Strukturierung von Requirements. Dies wird durch die Einordnung in semantische Schemata / Ontologien und durch die Bereitstellung semantischer Patterns und Taxonomien erreicht. Zur Entwicklung einer Requirements-Engineering Ontologie werden zunächst für das Requirements-Engineering zentrale Konzepte wie z.B. Requirement, Feature, Bug, Stakeholder und Developer definiert. In einem nächsten Schritt werden diese durch wichtige Attribute, wie z.B. Priorität, Kosten/Aufwand, Nutzen, Safety, Reliability und Usability, charakterisiert. Das entstandene Modell wird daraufhin bzgl. bereits existierenden Ontologien (wie z.B. FOAF/DOAP, vCard, iCal) ausgerichtet.

4.2 Prozess- und Interaktionsmodell

Ein wesentlicher Ansatzpunkt zur Definition von Requirements für evolutionäre System-(weiter-)entwicklung ist die Erfassung anforderungsrelevanter Äußerungen einer sehr großen Anzahl von Anwendern und Endbenutzern. Um diesen Prozess generisch abzubilden und domainspezifisch konfigurieren zu können ist die Entwicklung eines Prozess- und Interaktionsmodells geplant. Dazu werden alle

notwendigen Prozessschritte beim Requirements-Management identifiziert unter besonderer Berücksichtigung der Elicitation, des Modelling, der Analyse, des Agreeing und der Evolution von Requirements. Für die Motivation von Endanwendern und deren konkrete Interaktion mit den requirements-relevanten Komponenten müssen geeignete, insbesondere leicht nutzbare Interaktionsmöglichkeiten bereitgestellt werden.

4.3 Requirements-Elicitation und -Strukturierung

Der zentrale Aspekt der Requirements-Elicitation und -Strukturierung wird auf drei verschiedene Arten unterstützt:

- Moderation der Requirements-Elicitation,
- Analyse textuell beschriebener Anforderungen,
- Anwenderrückkopplung (Feedback) und Review

Zur Moderation der Requirementselicitation werden multimodale Eingabe- und Interaktionsmöglichkeiten bereit gestellt, die insbesondere die Aufnahme von Ideen, Konzepten und informellen Requirements-Beschreibungen in Gruppensitzungen ermöglichen und durch einen moderierten Prozess unterstützen.

Zur Analyse textuell beschriebener Anforderungen werden textuell gesammelte Anforderungen nach verschiedenen Gesichtspunkten bzgl. den, für das semantische Requirementsmodell entwickelten Taxonomien, kategorisiert, strukturiert und annotiert. Dabei kommen Basismechanismen zur automatisierten Extraktion von Themenstrukturen, zur statistischen Auswertung großer Textmengen bzgl. Begriffsassoziationen, sowie Metriken und Verfahren zur Bestimmung von Ähnlichkeiten zwischen Requirements zur Aggregation zum Einsatz.

Bereits beschriebene Anforderungen sowie deren Kategorisierungen und Annotationen können mittels Feedback und Review kommentiert und nach verschiedenen Gesichtspunkten bewertet werden. Protokollierung statistischer Daten wie z. B. Abfragehäufigkeit oder Änderungshäufigkeit, ebenso, wie eine strukturierte Anwenderrückkopplung z.B. über Nutzen/Wichtigkeit, Safety/Reliability, Usability sollen dabei einfließen.

4.4 SoftWiki-Framework

Alle entwickelten Verfahren und Methoden zur Akquisition, semantischen Strukturierung und zum Management von Requirements werden durch ein intuitiv nutzbares, webbasiertes Werkzeug unterstützt. Dieses soll besonders leicht installierbar sowie interoperabel mit bestehenden Softwaresystemen sein und die im Prozess- und Interaktionsmodell spezifizierten Prozesse auf der Basis des semantischen Requirementsmodells organisieren. Basierend auf Semantic-Web Technologien und

Community-Content-Management (Wiki-) Konzepten wird eine webbasierte Editierbarkeit und umfassende Versionierung aller Änderungen gewährleistet.

Existierende Requirements-Engineering Systeme werden mittels standardisierter Datenaustauschs- und Zugriffsformate (Adapter) mit dem entwickelten SoftWiki-Framework interagieren können. Darüber hinaus soll, das bei der Anforderungsanalyse gewonnene „Wissen“ anderen Phasen der Softwareentwicklung strukturiert zur Verfügung gestellt und somit der Softwarelebenszyklus einen Informations- bzw. „Wissensfluss“ begleitet werden.

5. Technologische Umsetzung des SoftWiki-Konzepts

Im Bereich des Community-Content- und Knowledge- Managements erfreut sich das Wiki-Konzept, des kollaborativen Editierens von Texten, zunehmender Popularität. In Ergänzung dazu soll das SoftWiki-Konzept die webbasierte, semantische Annotation von textuell beschriebenen Anforderungen und darüber hinaus die webbasierte, kollaborative Entwicklung von Anforderungs-Taxonomien, -Ontologien und -Wissensbasen und die daraus erwachsende Nutzung neuer Möglichkeiten des Peer-Review und Provenance-Tracking ermöglichen. Dabei wird die Wiki-Philosophie „Making it easy to fix mistakes, rather than making it hard to make them.“ auf kollaboratives Requirements-Engineering übertragen und ermöglicht damit eine iterative Verfeinerung, Annotierung und Strukturierung von Anforderungen und deren Verwendung im gesamten Softwarelebenszyklus.

Konkret erreicht wird dies durch:

- intuitive webbasierte Editierbarkeit von Anforderungsklassen (Schemata) und Anforderungen (Instanzen),
- Versionierung mit der Möglichkeit beliebige Änderungen transparent und selektiv rückgängig zu machen,
- Unterstützung der Evolution der Anforderungsklassen durch Ratgeber und Assistenten,
- Ermöglichung verschiedener Sichten auf die Inhalte,
- Weltweite Referenzierung und Syndikation (z. B. Wiederverwendung bereits definierter Anforderungsklassen),
- Semantische Suche unter Berücksichtigung der terminologischen Klassifizierungen,
- Automatische Klassifizierung neuer Requirements, Überprüfung der terminologischen Konsistenz.

Anderen SoftWiki-Nutzern soll ermöglicht werden, dediziert auf bestimmte terminologische Strukturierungen Bezug zu nehmen bzw. diese zu modifizieren oder entsprechend dem Peer-Review-Prinzip zu kommentieren und bewerten. Höchstes Ziel

des SoftWiki-Konzeptes ist dabei die Minimierung des Aufwands bei der Spezifikation von Anforderungen und deren späterer Modifikation und damit die Minimierung der Prozesskosten.

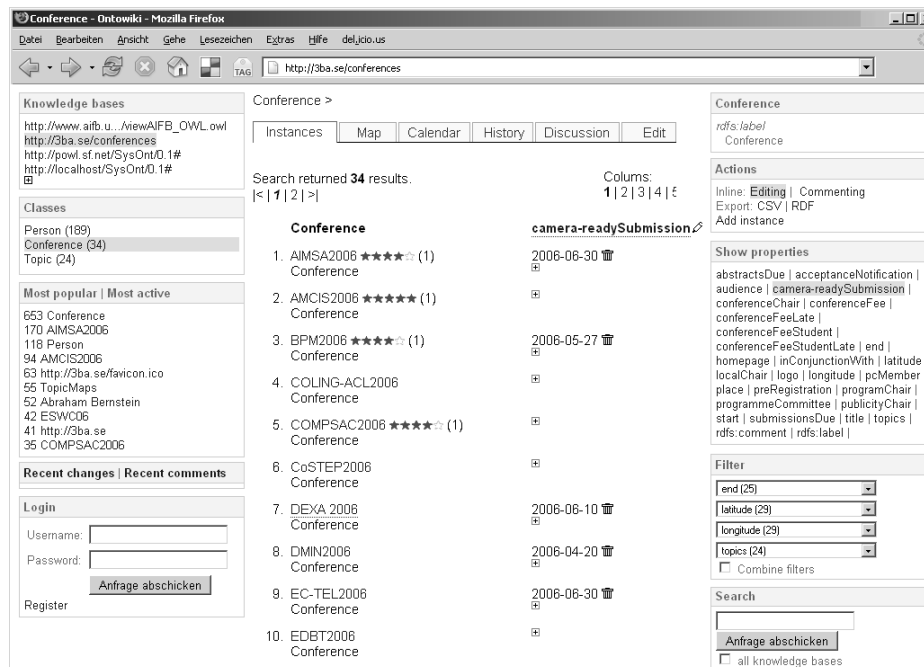


Abbildung 3: SoftWiki Prototyp (http://3ba.se)

Der Prototyp unterstützt das Softwiki-Konzept nur generisch. In Abbildung 3 wird dies am Beispiel einer Konferenz-Datenbank gezeigt. Eine Anpassung an das Requirements-Engineering und deren Evaluation ist derzeit Gegenstand der Entwicklung. Das Nutzerinterface ist komplett web-basiert und ordnet wichtige Elemente zur Darstellung, Navigation und zum Editieren dreispaltig an. Die linke Spalte des Web-Nutzerinterfaces gibt einen Überblick über vorhandene Wissensbasen bzw. Ontologien im System. Für die ausgewählte Ontologie wird die Klassenhierarchie dargestellt. Des Weiteren wird ein Überblick über statistische Daten, wie z.B. die populärsten Inhalte oder die aktivsten Teilnehmer/Autoren, gegeben.

Im zentral angeordneten Hauptansichtsbereich werden Nutzern verschiedene Sichten auf die zum aktuellen Navigationskontext gehörenden Objekte geboten. Ist eine Ontologie (oder eine Klasse) innerhalb einer Ontologie ausgewählt, enthält der Hauptansichtsbereich Übersichtsdarstellungen zu den in der Ontologie (oder Klasse) enthaltenen Instanzen. Beim Vorhandensein von räumlichen bzw. zeitlichen Informationen zu den Instanzdaten kann zu einer karten- oder kalendarischen Ansicht der entsprechenden Instanzen gewechselt werden. Ist eine konkrete Instanz ausgewählt, so werden im Hauptansichtsbereich alle der Instanz zugeordneten Eigenschaften dargestellt. Zusätzliche Eigenschaften lassen sich mittels interaktiver, dynamisch

nachgeladenen Editierwidgets mit minimalem Aufwand hinzufügen. Alle Änderungen werden in einer Änderungsansicht zusammengefasst, Diskussionsbeiträge zur Instanz oder einzelnen Instanzeigenschaften in einer Diskussionsansicht. Ein bzgl. der zugehörigen Klassendefinition dynamisch generiertes Formular ermöglicht das leichte Editieren aller Instanzeigenschaften.

Die rechte Spalte des Nutzerinterfaces enthält eine Volltextsuchfunktion und aufgrund der vorhandenen Instanzeigenschaften dynamisch generierte Filterfunktionen. Darüber hinaus lässt sich die Darstellung der Instanzübersicht in Bezug auf darzustellende Instanzeigenschaften, Anzahl von Spalten, Anzahl der Instanzen pro Seite usw. anpassen. Zur Unterstützung des Gemeinschaftsaspektes lassen sich einzelne Instanzen mit Bewertungen versehen.

Der beschriebene SoftWiki-Prototyp wurde auf der Basis des Semantic Web Anwendungsentwicklungs-Frameworks Powl erstellt. Er ist Open-Source und kann als Bestandteil von Powl unter <http://powl.sf.net> heruntergeladen werden. Unter <http://3ba.se> findet sich eine Demonstrationsinstallation mit verschiedenen Beispielwissensbasen.

6. Zusammenfassung

Ziel des SoftWiki-Konzepts ist die Entwicklung einer agilen Requirements-Akquisitions- und -Management Methode und deren Implementierung in einem Werkzeug. Überaus populäre Konzepte des Community-Content-Managements (WikiWikiWebs und Weblogs) werden dazu mit Methoden der semantischen Strukturierung und Wissensrepräsentation (Semantic-Web-Initiative W3C) verknüpft. Damit wird eine effektive Unterstützung der Gewinnung und Erfassung nicht formal beschriebener Anforderungen einer sehr großen Zahl, teilweise anonymer Anwender, wie sie beispielsweise im Kontext großer E-Government- bzw. B2C-Anwendungen zu finden sind, erreicht.

Während in den bekannten Methoden der agilen Softwareentwicklung Wissensmanagement implizit durch personelle und räumliche Konzentration erfolgt [ScBe01][Co04][BeAn04], verfolgt SoftWiki für die Integration großer Stakeholdergruppen eine explizite technische Unterstützung des Wissensmanagements im Requirements-Engineering. Ähnlich den Klassen von Softwareprojekten mit großer Anzahl verteilter Stakeholder, gibt es auch Softwareprojekte mit einer großen Anzahl verteilter Entwickler, wie z.B. Open Source Projekte. Nach einer erfolgreichen Erprobung des SoftWiki-Konzepts können die darin enthaltenen Methoden auch in anderen Softwareentwicklungsphasen angewandt werden. Bereits die semantische Strukturierung der Anforderungen ermöglicht deren Wiederverwendung auch in

anderen Entwicklungsphasen, wie z. B. der Dokumentation, dem Test (Bug-Tracking), dem Deployment und der Konfiguration und dem Support.

Das SoftWiki-Konzept basiert auf sehr aktuellen Technologien, welche noch in einem frühen Reifestadium sind. Es können daher vor allem noch keine Aussagen über die Skalierbarkeit des Konzepts getroffen werden. Eine prototypische Implementierung des Konzeptes wird im Rahmen eines Verbundprojektes durchgeführt.

7. Danksagung

Wir bedanken uns bei Prof. Klaus Pohl und Prof. Jürgen Ziegler von der Universität Essen-Duisburg, Dr. Frank Schönefeld von T-Systems, Dr. Mariele Hagen von ProDV, Herrn Andreas Sczepansky von QA-Systems sowie Herrn Veli Velioglu von ISA Informationssysteme für die Beiträge, Anmerkungen und Diskussionen zum vorliegenden Artikel.

Literaturverzeichnis

- [YuMy98] Yu, E.; Mylopoulos, J.: Why Goal-Oriented Requirements Engineering. In: Proc. of the 4th Int. Workshop on Requirements Engineering: Foundations of Software Quality (8-9 June, Pisa, Italy). Presses Universitaires de Namur, 1998, S. 15-22.
- [La01] v. Lamsweerde, C.: Goal-Oriented Requirements Engineering: A Guided Tour. In: Proc. of the RE'01, 5th Int. Symposium on Requirements Engineering, Toronto, August 2001, S. 249-263.
- [Yu97] Yu, E.: Why Agent-Oriented Requirements Engineering. In: Proc. of the 3rd Int. Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 97), Barcelona, Catalonia, Spain; 1997, S. 171-183.
- [GRL05] New Draft Recommendation Z.151 - Goal-Oriented Requirement Language (GRL). Version 3.0, Sept. 2003. http://www.usecasemaps.org/urn/z_151-ver3_0.zip
- [OI04] Olsson, T.; Denger, C.; Koenig, T.; Eisenbarth, M.; Schmid, K.: Ein Referenzmodell für Anforderungsspezifikationen. Treffen der Fachgruppe 2.1.6. „Requirements Engineering“. Kaiserslautern, 25./26.11, 2004 In: Softwaretechnik-Trends, (24)4, 2004
- [BeAn04] Beck, K. & Andres, C. : Extreme Programming Explained: Embrace Change, Second Edition, Addison Wesley Professional, 2005.
- [LeCu01] Bo Leuf and Ward Cunningham. The Wiki Way: Collaboration and Sharing on the Internet. Addison-Wesley Professional, 2001.

- [Be01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic Web. *Scientific American*, 284(5):34{43, May 2001.
- [Be04] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. Owl web ontology language reference. W3C Recommendation (<http://www.w3.org/TR/owl-ref/>), 2004.
- [BrMi05] Dan Brickley and Libby Miller. Foaf vocabulary specification. <http://xmlns.com/foaf/0.1/>, 2005.
- [Ho98] Horrocks. The FaCT system. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International ConferenceTableaux'98*, number 1397 in *Lecture Notes in Arti- cial Intelligence*, pages 307{312. Springer-Verlag, May 1998.
- [St94] Streitz, N. A., Geissler, J., Haake, J. & Hol, J. (1994). DOLPHIN: integrated meeting support across local and remote desktop environments and LiveBoards. *Proc. CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 345-358
- [Ga03] Gaßner, K. S. (2003): *Diskussionen als Szenario zur Ko-Konstruktion von Wissen mit visuellen Sprachen*. Dissertation: Fachbereich Elektrotechnik. Universität Duisburg.
- [DiBe04] DiMicco, J. M. & Bender, W. (2004). Second messenger: increasing the visibility of minority viewpoints with a face-to-face collaboration tool. *Proc. International Conference on Intelligent User Interfaces*, Funchal, Madeira, Portugal, 232-234
- [Kr05] Krötzsch, M.; Vrandečić, D. & Völkel, M. (2005), *Wikipedia and the Semantic Web - The Missing Links*, in Jakob Voss & Andrew Lih, ed., 'Proceedings of Wikimania 2005, Frankfurt, Germany'.
- [Au05] Aumüller, D.: SHAWN: Structure Helps a Wiki Navigate, in *Proceedings of the BTW-Workshop "WebDB Meets IR"*, 2005.
- [Ta04] Tazzoli, R.; Castagna, P. & Campanini, S.E.: 'Towards a Semantic Wiki Wiki Web', Poster at *Third International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan, 2004.
- [ScBe01] Schwaber, K. & Beedle, M.: *Agile Software Development with Scrum*, Prentice Hall, 2001.
- [Co04] Cockburn, A.: *Crystal Clear*, Addison-Wesley Professional, 2004.
- [W3C01] Koivunen, Miller: *W3C Semantic Web Activity*, 2001. <http://www.w3.org/2001/12/semweb-fin/w3csw>