

**Einsatzpotentiale von Wikis in der Softwareentwicklung am  
Beispiel von Requirements Engineering und Traceability  
Management**

**Michael Geisser and Hans-Jörg Happel and  
Tobias Hildenbrand and Stefan Seedorf**

Working Paper 7/2007  
March 2007

**Working Papers in Information Systems**

---

**University of Mannheim**  
Department of Information Systems 1  
D-68131 Mannheim/Germany  
Phone +49 621 1811691, Fax +49 621 1811692  
E-Mail: wifo1@uni-mannheim.de  
Internet: <http://www.bwl.uni-mannheim.de/wifo1>

# Einsatzpotentiale von Wikis in der Softwareentwicklung am Beispiel von Requirements Engineering und Traceability Management

Michael Geisser<sup>1</sup>, Hans-Jörg Happel<sup>2</sup>, Tobias Hildenbrand<sup>1</sup> und Stefan Seedorf<sup>3</sup>

<sup>1</sup>: Universität Mannheim, Lehrstuhl für ABWL und Wirtschaftsinformatik.  
{geisser|hildenbrand}@wi.uni-mannheim.de

<sup>2</sup>: Forschungszentrum Informatik, Information Process Engineering  
happel@fzi.de

<sup>3</sup>: Universität Mannheim, Lehrstuhl für Wirtschaftsinformatik III  
seedorf@uni-mannheim.de

## *1 Einleitung*

Wiki-Systeme sind günstige und einfach bedienbare Werkzeuge zur kollaborativen Erfassung und Verteilung von Wissen. Aufgrund der erfolgreichen Nutzung in vielen Internet-Gemeinschaften werden sie zunehmend auch im professionellen Unternehmensumfeld eingesetzt [MaWY06]. Insbesondere Softwareentwicklungsprojekte stellen ein klassisches Einsatzgebiet für Wikis dar. Die Anwendungsfelder reichen dabei von der Ideensammlung und Dokumentation über Fehlermanagement bis hin zur Projektkoordination. Damit werden jedoch nicht alle Potenziale für eine umfassende Unterstützung von Softwareentwicklungsprozessen ausgeschöpft. Es stellt sich unter anderem die Frage, wie die Kernaktivitäten im Software-Engineering (SE) besser als bisher unterstützt werden können.

Dieser Beitrag zeigt die Einsatzpotenziale von Wikis in ausgewählten Aktivitäten des SE auf. Nach einer kurzen Vorstellung der Wiki-Technologie betrachten wir insbesondere den Nutzen von Wikis in der Softwareentwicklung. Im Hauptteil dieses Beitrags werden mit Requirements Engineering (RE) sowie Traceability und Rationale Management (TRM) zwei konkrete Anwendungsszenarien vorgestellt. Im Rahmen des RE, der initialen Phase eines Softwareentwicklungsprojekts, werden die Anforderungen an das geplante System erhoben, analysiert, spezifiziert und validiert. Wikis stellen hierbei eine leichtgewichtige und agile Alternative zu einer Reihe anderer kommerzieller und teilweise sehr komplexer Lösungen dar. Im RE und in den folgenden Phasen der Softwareerstellung müssen neben den Abhängigkeitsbeziehungen der Anforderungen untereinander und zwischen den daraus entstehenden Artefakten auch die zugrunde liegenden Entscheidungsprozesse (engl. "Rationale") der beteiligten Personen festgehalten und nachvollziehbar gemacht werden. Zur Erfassung und Vernetzung dieser Informationen bedarf es eines systematischen TRM-Prozesses, für dessen Realisierung wiederum Wiki-

Systeme aufgrund ihrer einfachen Benutzbarkeit bei gleichzeitig hoher semantischer Mächtigkeit sehr gut geeignet erscheinen.

Nach einer Vorstellung dieser beiden konkreten Einsatzmöglichkeiten folgt ein Ausblick auf so genannte „Semantische Wikis“, die traditionelle Wikis um die Möglichkeit einer maschinen-interpretierbaren Auszeichnung von Inhalten erweitern und somit zu einer integrierten Prozessunterstützung beitragen können.

Die hier vorgestellten Methoden und Werkzeuge wurden im Rahmen des Forschungsprojekts CollaBaWue entwickelt. Die Anwendungsbeispiele basieren auf dem Wiki-System der kollaborativen Softwareentwicklungsplattform CodeBeamer der Firma Inland, sind aber im Wesentlichen auf die meisten anderen Wiki-Systeme übertragbar.

## *2 Vorstellung der Wiki-Technologie*

Als "Wikis" (hawaiianisch für „schnell“) werden Softwareprogramme bezeichnet, die sich besonders leicht zur web-basierten Zusammenarbeit von mehreren Personen nutzen lassen. Dabei unterscheidet man zwischen der eigentlichen Software ("Wiki-Engine") und der Webseite ("Wiki", "Wiki-Installation"). Beispielsweise verwendet die bekannte Online-Enzyklopädie Wikipedia die Wiki-Engine "MediaWiki". Während die meisten Wiki-Engines als Open Source-Software verfügbar sind, kommen in letzter Zeit verstärkt kommerzielle Produkte sowie Hosting/ASP Anbieter auf den Markt (z.B. SocialText oder JotSpot).

Grundsätzlich besteht ein Wiki aus einer Vielzahl eindeutig benannter Webseiten, die untereinander verlinkt sind – ähnlich dem World Wide Web, das sich ebenfalls aus verlinkten Dokumenten zusammensetzt. Im Gegensatz dazu lassen sich Wiki-Seiten jedoch direkt im Browser erstellen und ändern, worin sie klassischen Web Content Management Systemen (WCMS) ähneln. Sie unterscheiden sich allerdings darin, dass weder feste Inhaltskategorien noch Einschränkungen bei den Nutzerrechten vorgegeben werden. In der Regel ist es jedem Benutzer möglich, die Inhalte eines Wikis zu editieren. Die Wiki-Engine pflegt jedoch eine Änderungshistorie, über die eine alte Version wiederhergestellt werden kann. Soziale Kontrolle ersetzt folglich das aufwändige Rechtemanagement in WCMS [EbG105].

Da ihre Basisfunktionen ähnlich einfach zu erlernen ist wie die gängiger web-basierter E-Mail-Dienste, weisen Wikis eine niedrige Einstiegshürde auf. Obwohl dieser Vorteil schnelle Erfolge und Nutzerzufriedenheit bei der Bearbeitung von textbasiertem Wissen verspricht, bieten sie nur geringe Hilfestellung bei der Strukturierung dieses Wissens. Deshalb neigen Wikis mit zunehmender Nutzung zur Unübersichtlichkeit und Ausuferung der Inhalte [MaWY06]. An dieser Schwachstelle setzen Semantische Wikis an, die in Abschnitt 4 genauer vorgestellt werden.

### 3 Einsatzgebiete von Wikis im Software-Engineering

Zunächst erfolgt eine kurze Übersicht der am weitesten verbreiteten Einsatzgebiete von Wikis im SE. Daraufhin werden das RE (Abschnitt 3.2) und TRM (Abschnitt 3.3) als Schwerpunkte betrachtet.

#### 3.1 Allgemeine Einsatzgebiete

Die Wiki-Technologie hat ihren Ursprung in der technikaffinen Gemeinde der Softwareentwickler. So widmet sich das "Ur-Wiki" von Ward Cunningham<sup>1</sup> seit 1995 dem Zusammentragen von Entwurfsmustern ("Patterns") in Softwaresystemen. Seitdem ist die Softwareentwicklung eines der dominanten Anwendungsgebiete von Wiki-Technologie geblieben [Lour06]. Häufig werden Wikis dabei von einer kleinen Nutzergruppe eingerichtet und dann von einer immer größeren Anzahl von Mitarbeitern genutzt. Dabei hilft vor allem die einfache Verfügbarkeit und Installation von Wiki-Engines.

Neben Open Source-Entwicklergemeinschaften und kleineren Firmen nutzen auch große Softwarefirmen wie SAP, Novell oder Yahoo Wiki-Technologie. Anwendungsgebiete sind dabei allgemeiner Wissensaustausch, technische Dokumentation, Qualitäts- und Prozessmanagement, Releaseplanung und Fehlerverfolgung [BaMe05, MaWY06, TWIK06]. Dabei ist zu beobachten, dass Wikis mit ihrer Kernfunktionalität bei spezifischen, gut strukturierbaren Inhalten häufig an ihre Grenzen stoßen. So gibt es bereits Erweiterungen, die spezifische Funktionalität für einzelne Problemfelder wie Quellcode-Dokumentation [AgDa05, JoSW05] oder Fehlerverfolgung [Edge06] ergänzen. Im Folgenden betrachten wir speziell die Nutzung von Wikis im RE und im TRM.

#### 3.2 Wikis im Requirements Engineering

Das RE, d.h. die systematische Erhebung, Analyse, Spezifikation, Validierung und Verwaltung der Anforderungen an die Software, wird herkömmlich entweder mit schwergewichtigen RE-Werkzeugen oder sogar ausschließlich mittels herkömmlicher Büro-Software erledigt. Traditionelle RE-Werkzeuge, wie z.B. RequisitePro und DOORS, sowie Office-Produkte sind zwar weit verbreitet, originär aber nicht für eine internetbasierte Umgebung konzipiert worden. Dies hat u.a. Performance-Schwierigkeiten und ineffiziente Prozesse zur Folge. Verteiltes RE mittels Büro-Software kann für größere Softwareprojekte allenfalls als

<sup>1</sup> <http://www.c2.com/>

Behelfslösung betrachtet werden. Hierbei werden Dokumente häufig lediglich per E-Mail verteilt und nicht zentral versioniert.

Obwohl verteilten Ansätzen im RE ein hohes Erfolgspotential attestiert wird, existieren bis dato keine durchgängigen Methoden für ein verteiltes RE. Daher soll nun mit DisIRE (**D**istributed **I**nternet Based **R**equirements **E**ngineering) eine Wiki-gestützte Methode zum verteilten, internetbasierten RE mitsamt einer spezifischen Werkzeugunterstützung, die in existierende Entwicklungsplattformen integriert werden kann, vorgestellt werden. DisIRE führt bewährte Ansätze im RE zusammen und erweitert diese, so dass die Methode auf einem soliden theoretischen und empirisch validierten Gerüst basiert<sup>2</sup>. Zudem bietet DisIRE zum ersten Mal eine durchgängige RE-Methode für ein verteiltes Umfeld, in der Abhängigkeiten von Anforderungen explizit berücksichtigt werden. Mit dieser Methode wird bspw. ein Systemhaus in die Lage versetzt, Anforderungen größtenteils verteilt und gleichzeitig systematisch mit den Kunden zu erheben, zu analysieren, auszuwählen, zu spezifizieren und zu validieren. Ein Moderator führt hierbei alle Beteiligten durch den im Folgenden beschriebenen Prozess.

### 3.2.1 Anforderungserhebung und -analyse

DisIRE setzt im RE direkt nach der Machbarkeitsstudie an. Zu diesem Zeitpunkt gibt es neben einer Vision des geplanten Systems auch die Sicherheit, dass es sich hierbei prinzipiell um ein realisierbares Projekt handelt. Als erster Schritt erfolgt in der Regel die Anforderungserhebung und -analyse, deren Ablauf in diesem Abschnitt beschrieben wird.

*Initiales Treffen:* Da auch bei verteiltem Arbeiten ein persönliches Kennenlernen aller Beteiligten nachweisbar zu besseren Ergebnissen führt (vgl. [GeHi06]), erfolgt zu Beginn ein Treffen mit möglichst vielen Beteiligten, mindestens jedoch mit einem Vertreter eines jeden Standorts und jeder betroffenen Organisationseinheit. Hierbei wird sichergestellt, dass die Anwesenden ein einheitliches Bild über das geplante System bekommen. Anschließend wird der folgende Prozess vorgestellt.

*Asynchrone Anforderungserhebung und -analyse:* Nach dem ersten Treffen werden die Anforderungen asynchron erhoben und analysiert. Tab. 1 gibt Auskunft über die hierbei teilnehmenden Rollen und deren Aufgaben. Sind alle Anforderungen kommentiert, konsolidiert sowie geprüft, und zeichnet sich zudem ab, dass keine weiteren Kommentare oder Anforderungen übermittelt werden, so können die Anforderungen durch den Moderator fixiert werden.

<sup>2</sup> Zu diesen Ansätzen gehören EasyWinWin [Grün03], QuantitativeWinWin [RuEP03] sowie der Cost-Value-Approach [KaRy97]. Vorarbeiten zu DisIRE sind in [GeHi06] beschrieben.

Rollen	Aufgaben
Interessenvertreter auf Kundenseite	<ul style="list-style-type: none"> <li>• Anforderungen übermitteln und kommentieren</li> <li>• Identifikation von erklärungsbedürftigen Fachtermini</li> <li>• Definition von erklärungsbedürftigen Fachtermini</li> <li>• Vision erweitern und präzisieren</li> </ul>
Moderator	<ul style="list-style-type: none"> <li>• Präzisierung forcieren</li> <li>• Anforderungen konsolidieren und kategorisieren</li> <li>• Identifikation von erklärungsbedürftigen Fachtermini</li> <li>• Vision erweitern und präzisieren</li> </ul>
Softwaretechniker	<ul style="list-style-type: none"> <li>• Anforderungen kommentieren</li> <li>• Machbarkeit von Anforderungen überprüfen</li> <li>• Identifikation von erklärungsbedürftigen Fachtermini</li> </ul>

**Tabelle 1: Rollen mit zugehörigen Aufgaben bei der Anforderungserhebung und -analyse**

Ein Wiki eignet sich hierfür besonders gut: Für jede initiale Anforderung wird eine eigene Wiki-Seite erstellt, auf der die zugehörigen Kommentare erfolgen. Letztere können dabei direkt in die Beschreibung der Anforderung konsolidiert werden. Ein integriertes Versionierungssystem stellt dabei sicher, dass die Änderungen auf jeder Seite nachvollziehbar bleiben. Außerdem werden Artikel oder einzelne Bereiche während der Editierung gesperrt, so dass keine Konflikte entstehen können. Ein weiterer Vorteil von Wikis ist die Möglichkeit, Glossare integrativ zu verwenden: Fachtermini, die bei der Beschreibung von Anforderungen verwendet werden und einer Erklärung bedürfen, können mittels Wiki-Links markiert und in einem Glossar beschrieben werden (vgl. Abb. 1). Auch die weiteren asynchronen Aktivitäten werden vollständig und sehr flexibel durch ein Wiki unterstützt. Ein solches Wiki ist z.B. in der kollaborativen Softwareentwicklungsplattform CodeBeamer der Firma Intland Software<sup>3</sup> integriert. CodeBeamer ist eine auf J2EE basierende Plattform mit integrierten Funktionalitäten zum Dokumenten-, Änderungs- und Workflow-Management sowie Issue Tracking und wurde im Rahmen des Forschungsprojekts CollaBaWü<sup>4</sup> verwendet, um den DisIRE-Prozess ganzheitlich zu unterstützen.

3 <http://www.intland.com/>

4 <http://www.collabawue.de/>



**Abb. 1: Markierung von Fachtermini in einem Wiki-System**

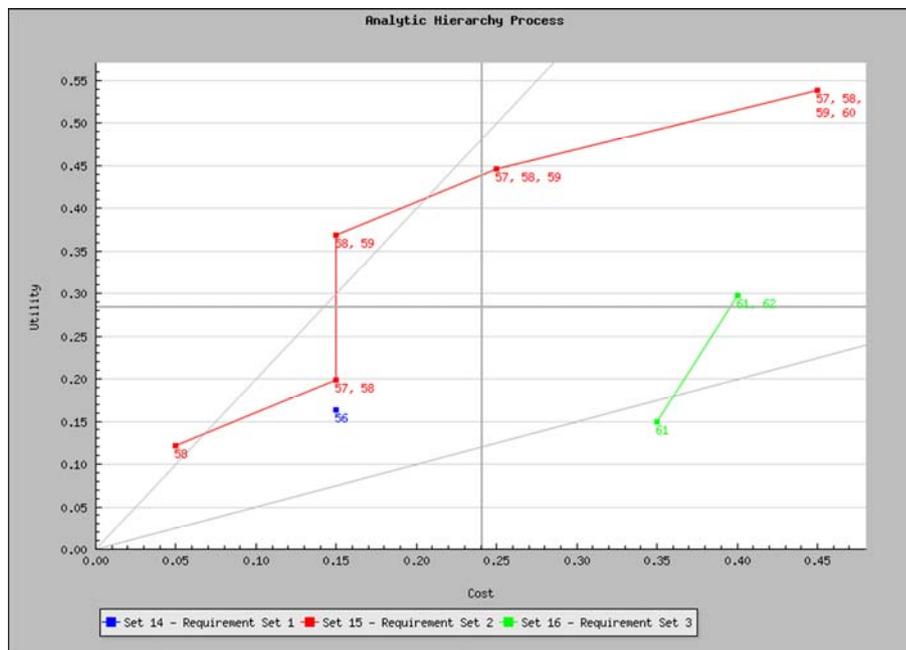
### 3.2.2 Anforderungsauswahl

Nach der Anforderungserhebung und -analyse liegt eine konsolidierte Liste mit Anforderungen vor, deren vollständige Umsetzung allerdings nicht immer ökonomisch rational ist. So können Anforderungen enthalten sein, deren Implementierung bei gleichzeitig relativ geringem Nutzen sehr aufwändig sein kann. Daher gilt es, jene Anforderungen auszuwählen, deren Umsetzung unter ökonomischen Gesichtspunkten sinnvoll erscheint. Die einzelnen Schritte der Anforderungsauswahl mit DisIRE werden im Folgenden kurz aufgezeigt.

*Anforderungssets bilden:* Da Anforderungen untereinander Abhängigkeiten aufweisen können, dürfen sie nicht unter Vernachlässigung dieser Abhängigkeiten miteinander verglichen werden. Daher werden abhängige Anforderungen zu so genannte Anforderungssets zusammengefasst, die mittels eines gerichteten Grafen dargestellt werden können (vgl. [GeHi06]).

*Kosten und Nutzen schätzen:* Nun können die Kosten und der Nutzen der Anforderungen geschätzt werden. Während die Softwaretechniker realistische Aufwandsschätzungen für die Anforderungen geben, schätzen die Interessenvertreter auf Kundenseite den Nutzen sowohl auf Ebene der Anforderungssets als auch innerhalb der Anforderungssets mittels des Analytic Hierarchy Process-Verfahrens, das auf paarweisen Vergleichen basiert [SaVa01].

*Grafische Analyse der Ergebnisse und Auswahl:* Nun werden die Ergebnisse unter Berücksichtigung der Abhängigkeiten grafisch dargestellt, indem ausgehend vom Wurzelement der gerichteten Grafen alle möglichen Kombinationen mit ihren aggregierten Nutzenwerten und Kosten in ein Diagramm eingetragen werden (vgl. Abb. 2). Auf dieser Basis erfolgt eine grafische Analyse und Auswahl der zu implementierenden Anforderungen.



**Abb. 2: Visualisierung der Nutzen/Kosten-Verhältnisse aller möglichen Kombinationen**

*Werkzeugseitige Umsetzung:* Um die eben vorgestellten Schritte der Anforderungsauswahl zu unterstützen, wurde an der Universität Mannheim mit ibere (vgl. [GeHi06]) ein Prototyp entwickelt, mit dem es möglich ist, Anforderungen von strukturierten Wiki-Seiten mittels eines Parsers zu importieren. Hierzu werden Meta-Informationen genutzt, welche im jeweiligen Wiki-System hinterlegt sein müssen. Nach dem Import erfolgen die eben beschriebenen Schritte der Anforderungsauswahl mit DisIRE, bevor die ausgewählten Anforderungen abschließend exportiert werden. Hierzu wird pro Anforderung ein Eintrag (Item) im Requirements Tracker von CodeBeamer angelegt.

### 3.2.3 Anforderungsspezifikation und -validierung

Im Anschluss an die Anforderungsauswahl sollten die ausgewählten Anforderungen spezifiziert werden, um einen ausreichenden Formalisierungsgrad zu erreichen. Werden wie bei DisIRE direkt Interessenvertreter auf der Kundenseite nach ihren Anforderungen gefragt, so handelt es sich meist um funktionale Anforderungen. Zur Spezifikation dieser Art von Anforderungen eignen sich

insbesondere Use Cases. Zur internetbasierten Spezifikation werden daher direkt im Requirements Tracker von CodeBeamer Strukturvorlagen (Templates) für diese Use Cases vorgehalten. Softwaretechniker können somit an jedem Arbeitsplatz mit Internetverbindung dieses Template zur Spezifikation der Anforderungen verwenden. Die so spezifizierten Anforderungen müssen abschließend noch von den Interessenvertretern auf Kundenseite validiert werden, um sicherzustellen, dass während der Spezifikation keine Informationen verloren gegangen und keine Missverständnisse aufgetreten sind. Hierfür wird auf die in den CodeBeamer Requirements Tracker integrierte Kommentar-Funktionalität zurückgegriffen.

Um sowohl die Spezifikationen als auch die zugehörigen Kommentare zu versionieren und eine parallele Bearbeitung zu ermöglichen, wurde das Wiki direkt in den Requirements Tracker und die Kommentarfunktionalität von CodeBeamer integriert. So können Wiki-Inhalte sowie andere Tracker Items und beliebige Artefakte aus CodeBeamer-Projekten bei der Kommentierung mittels Wiki-Links sehr agil und leicht referenziert werden.

### 3.3 Wikis im Traceability und Rationale Management

Auch in den weiteren Phasen der Softwareerstellung nach dem RE sollten die Abhängigkeitsbeziehungen der Anforderungen untereinander und zwischen den daraus entwickelten Artefakten sowie deren zugrunde liegenden Entscheidungsprozesse festgehalten werden (Traceability und Rationale Management). Dabei werden insbesondere die unterschiedlichen Abhängigkeiten sowie die genauen Gründe und Auswahlalternativen für Entscheidungen in den einzelnen Prozessschritten, bspw. Änderungen, verwaltet [DMMP06]. Diese Aufgabe gestaltet sich umso schwieriger, je verteilter die unterschiedlichen Beteiligten agieren [RaJa01]. Daher soll im Folgenden, ausgehend von der Spezifikation der Anforderungen im RE, mit „Wiki-TRaM“ ein Wiki-basierter TRM-Prozess zur Erfassung, Verwaltung, Visualisierung und Analyse dieser Informationen vorgestellt werden. Wiki-TRaM besteht aus den beiden Disziplinen „Erfassen und Verwalten“ sowie „Visualisierung und Analyse“, die in den folgenden Abschnitten näher erläutert werden.

#### 3.3.1 Erfassen und Verwalten von Traceability und Rationale-Informationen

Auch wenn Anforderungen bei DisIRE systematisch erhoben, analysiert und ausgewählt werden, ist die Spezifikation in den meisten Fällen noch nicht vollständig und stabil. Daher wird komplementär ein ebenso systematisches Änderungsmanagement benötigt. Voraussetzung hierfür ist die Nachvollziehbarkeit des Entstehungsprozesses der Anforderungen (Source Traceability) sowie der Abhängigkeiten der Anforderungen untereinander (Requirements Traceability) und

zu den daraus abgeleiteten Artefakten (Design Traceability) [Somm04]. Für das Anforderungs- und Änderungsmanagement im Rahmen des RE sind insbesondere die Abhängigkeiten zwischen den Artefakten in der Phase vor der eigentlichen Spezifikation („Prä-Spezifikationsphase“) von großer Bedeutung. Hierzu zählen unter anderem die Anforderungsbeschreibungen und Kommentare, die über die entsprechende Wiki-Seite mit den spezifizierten Anforderungen in einem Issue Tracker verlinkt sind. In Issue Trackern werden Aufgabenbeschreibungen und –stati verwaltet. Diese wiederum werden zusammen mit den Abhängigkeiten zwischen den spezifizierten Anforderungen, den Kosten- und Nutzenwerten sowie den Begründungen für die Auswahlentscheidungen im Issue Tracker von CodeBeamer abgelegt. Neben den Traceability und Rationale-Informationen kann somit auch der Umsetzungsstatus der einzelnen Anforderungen verwaltet und die entsprechenden Artefakte versioniert werden.

Im weiteren Verlauf eines Softwareprojekts gilt es in der Post-Spezifikationsphase die Beziehungen der spezifizierten Anforderungen zu den resultierenden Artefakten, wie z.B. Entwurfsmodellen, Quelltext und Testfällen, festzuhalten. Überdies sollen wiederum alle wesentlichen Entscheidungen samt deren Begründungen (Design Rationale) erfasst werden. Wikis bieten hierfür die entsprechende Funktionalität zur Vernetzung und Kommentierung der Inhalte.

Dies erleichtert u.a. die Anpassung, Wartung und Wiederverwendung des Systems und ermöglicht eine schnellere Einarbeitung neuer Projektmitarbeiter. Zu diesem Zweck werden auf der CodeBeamer-Plattform unterschiedliche Tracker zu den jeweiligen Aufgabenfeldern (RE, Architekturentwurf, Entwicklung etc.) verwendet und abhängige Aufgaben über deren Assoziationsmechanismus bzw. Wiki-Links verbunden. Mittels Assoziationen werden typisierte Beziehungen zwischen Tracker Items untereinander, beliebigen Artefakten sowie externen URLs festgehalten. Neben dem CodeBeamer-eigenen Dokumentenmanagementsystem können auch Artefakte, bspw. Quelltext, aus externen Repositories wie Subversion referenziert werden. Jede Änderung an einem Artefakt erfordert die Erstellung eines Kommentars bzw. bei SVN einen kommentierten „Commit“. Da hierbei wiederum die Wiki-Kommentare zum Einsatz kommen, können somit ebenfalls Beziehungen zu beliebigen CodeBeamer-Artefakten sowie externen Ressourcen erstellt werden. Mittels dieser Mechanismen (Assoziationen und Wiki-Links) werden somit Traceability und Rationale-Informationen über den gesamten Projektverlauf hinweg dokumentiert und vernetzt.

### 3.3.2 Visualisierung und Analyse von Traceability und Rationale-Informationen

Durch die detaillierte Erfassung der oben genannten Informationen entsteht ein heterogenes „Trace-Netzwerk“, das Beziehungen zwischen Prozessschritten (d.h. Tracker Items), Artefakten und beteiligten Personen beinhaltet. Zur übersichtlicheren Visualisierung und zur effektiveren Analyse dieser Traceability-Informationen wurde daher das Werkzeug „TraVis“ entwickelt (TraVis = Trace

Visualization). Es handelt sich dabei um eine Java-Applikation, die auf Basis der CodeBeamer Web-Schnittstelle auf die entsprechenden Basisinformationen zurückgreift und diese gemäß rollenbasierter Filter darstellt. Diese Filter erlauben es, verschiedene Sichten für bspw. Entwickler, Berater, Projektmanager etc. zu definieren. Dabei können sowohl die angezeigten Kategorien von Artefakten als auch die dargestellten Beziehungen variiert werden. Des Weiteren können einzelne Artefakte, Tracker-Einträge oder Benutzer samt deren unmittelbaren und mittelbaren Traceability-Netzwerk herausgegriffen werden. Auf weitere Details dieses Werkzeugs wird hier nicht weiter eingegangen, da es lediglich die Daten aus einem Wiki weiterverarbeitet und somit keine Social Software im engeren Sinn darstellt.

#### *4 Semantische Wikis*

Während herkömmliche Wikis sich gut zur kollaborativen Bearbeitung unstrukturierter Informationen eignen, sind sie bezüglich einer tieferen Strukturierung des darin enthaltenen Wissens begrenzt. Neben dem zuvor vorgestellten Ansatz, einzelne Wiki-Engines für die Bearbeitung bestimmter Arten strukturierter Daten zu erweitern, bieten so genannte „Semantic Wikis“ ein allgemeines Konzept für die Bearbeitung strukturierten Wissens [Onto06].

Semantische Wikis erweitern existierende Wiki-Ansätze mit Technologien des „Semantic Web“. Sie ermöglichen die formale Repräsentation von Wissen, anstatt es ausschließlich auf Basis eines Hypertext-Formats zu beschreiben. Analog versteht man unter dem "Semantic Web" [BeHL01] die Vision, Web-basierte Inhalte maschinen-interpretierbar zu beschreiben, damit „intelligente“ Werkzeuge den Menschen bei der Informationssuche und -Filterung besser als bisher unterstützen können. Als erste Schritte in diese Richtung können die automatische Aggregation von RSS-Feeds oder auch Web-APIs<sup>5</sup> angesehen werden.

Das World Wide Web Consortium (W3C) entwickelt unter dem Begriff „Semantic Web Technologien“ eine Reihe von Standards und Spezifikationen, um diese Vision realisieren zu können. Wichtige Elemente sind dabei Sprachen zur formalen Wissensrepräsentation, wie z.B. die Web Ontology Language (OWL) [W3C04]. Diese ermöglichen die strukturierte Beschreibung von Zusammenhängen zwischen den Konzepten einer Domäne. Ein gutes Beispiel hierfür ist das Projekt Semantic Wikipedia [VöKr06]. Während die Information, dass bspw. Berlin die Hauptstadt von Deutschland ist, für den Menschen aus den Wikipedia-Einträgen von "Berlin" und "Deutschland" klar hervorgeht, ist dies für Softwareprogramme nur sehr schwer erkennbar.

Da die vom Wikipedia-Projekt genutzte Software MediaWiki hier keine Hilfe bietet, erweitert das Modul SemanticMediaWiki die Syntax von Verknüpfungen um eine semantische Ebene. Die Verlinkung der Seite "Berlin" im "Deutschland"-

<sup>5</sup> siehe z.B.: <http://www.programmableweb.com>

Artikel kann dadurch um die Meta-Information "ist Hauptstadt" ergänzt werden. Somit lassen sich bei entsprechender Wissensanreicherung deklarative Anfragen wie z.B. eine Suche nach „allen Hauptstädten in Europa“ schnell formulieren und automatisiert beantworten.

Diese Erweiterung kollidiert nicht mit der normalen Wiki-Funktionalität sondern ist additiv. Die Funktion sozialer Software ist hier also nicht auf die reine Texterstellung beschränkt, sondern wird erweitert um die Möglichkeit schrittweise strukturiertes Wissen beizusteuern. Die Wikipedia könnte damit etwa zu einer mächtigen, abfragbaren Datenbank über das bisher in den Artikeln verborgene Wissen werden.

Gerade im SE können semantische Wikis nützlich sein. So ermöglichen sie eine schrittweise Formalisierung des zugrunde liegenden Wissens von den Phasen Anforderungsanalyse und Design bis hin zur Testphase. Insbesondere die Möglichkeit der Verknüpfung von Artefakten über verschiedene Phasen des Entwicklungsprozesses hinweg ("Traceability", vgl. Abschnitt 3.3) bieten bisher nur die integrierten Lösungen großer Hersteller.

Das kollaborativ erarbeitete, formalisierte Wissen ersetzt dabei nicht natürlichsprachige Erläuterungen sondern ergänzt diese. Für eine automatische Verarbeitung innerhalb oder außerhalb des Wikis stehen u.a. Schlussfolgerungssysteme, Regelsprachen oder deklarative Anfragesprachen zur Verfügung. Diese können z.B. für die dynamische Generierung von Inhalten oder zur Validierung der Modelldaten genutzt werden. Des Weiteren ist es möglich, die Modelldaten anderen Werkzeugen zur Verfügung zu stellen, z.B. um Quellcode aus einem im Wiki spezifizierten Domänenmodell zu generieren.

## *5 Zusammenfassung und Ausblick*

Wie in diesem Beitrag dargelegt wurde, kann die „Social Software“ Wiki (vgl. [Bäch06]) nicht nur in klassischen Anwendungsfeldern der Softwareentwicklung, z.B. zur verteilten Dokumentation, einen signifikanten Nutzen bringen. Im Zuge der fortwährenden Evolution der Wiki-Technologie und der Entwicklung neuer Methoden können davon auch weitere Einsatzgebiete im SE profitieren. Exemplarisch wurden hierbei mit Requirements Engineering sowie Traceability und Rationale Management zwei konkrete Anwendungsszenarien herausgegriffen.

Diese beiden Anwendungsfälle und deren Wiki-basierte Implementierungen verdeutlichen, dass Wikis keineswegs auf den Open Source-Bereich beschränkt sind, sondern auch einen interessanten und flexiblen Lösungsansatz zur Unterstützung von Unternehmensprozessen darstellen. Im Open Source-Umfeld wurden Wikis bisher vor allem als „agiles“ Werkzeug für das Wissensmanagement und die asynchrone Zusammenarbeit eingesetzt. In Zukunft sollten neben RE und TRM weitere Einsatzgebiete in der Softwareentwicklung erschlossen werden, da im Zuge einer Spezialisierung der Rollen im Entwicklungsprozess immer mehr Informationen verteilt gehalten werden und automatisch verarbeitet werden müssen.

Dazu ist es jedoch auch erforderlich, die noch existierenden Schwachpunkte von klassischen Wikis in Bezug auf die maschinen-interpretierbare Strukturierung des enthaltenen Wissens zu kompensieren. In diesem Zusammenhang ermöglichen es Semantische Wikis, die informellen Wiki-Inhalte schrittweise aufzuwerten und mit formal spezifiziertem Wissen zu verknüpfen. Ein nahtloser Übergang zwischen den fachlich- und technisch-orientierten Spezifikationen kann dabei letztlich zu einer besseren Kommunikation zwischen den verschiedenen Prozessbeteiligten beitragen.

## 6 Literaturverzeichnis

- [AgDa05] Aguiar, A.; David, G.: WikiWiki weaving heterogeneous software artifact. In: Proceedings of the 2005 International Symposium on Wikis, San Diego, CA, 2005, S. 67-74.
- [Bäch06] Bächle, M.: Social Software. In: Informatik Spektrum, 2006, 29, S. 121-124.
- [BaMe05] Bachmann, F.; Merson, P.: Experience Using the Web-Based Tool Wiki for Architecture Documentation. Technical Note CMU/SEI-2005-TN-041. September 2005.
- [BeHL01] Berners-Lee, T.; Hendler J.; Lassila, O.: The Semantic Web. In: Scientific American, 284, 5, 2001.
- [DeRe05] Decker, B.; Rech, J.; Ras, E.; Klein, B.; Hoecht, C.: Self-organized Reuse of Software Engineering Knowledge supported by Semantic Wikis. In: Proceedings of the Workshop on Semantic Web Enabled Software Engineering (SWESE). November 2005.
- [DMMP06] Dutoit, A.H.; McCall, R.; Mistrik, I.; Paech, B. (Hrsg.) Rationale Management in Software Engineering, Springer Verlag, 2006.
- [EbGI05] Ebersbach, A.; Glaser, M.: Wiki. In: Informatik Spektrum, (28), 2005, S. 131-135
- [Edge06] Edgewell.org: The Trac User and Administration Guide, URL: <http://trac.edgewall.org/wiki/TracGuide> (26.09.2006), 2006.
- [GeHi06] Geisser, M.; Hildenbrand, T.: A Method for Collaborative Requirements Elicitation and Decision-Supported Requirements Analysis. In: Ochoa, S.F. und Roman, G.-C. (eds): IFIP International Federation for Information Processing, 2006 (219), Advanced Software Engineering: Expanding the Frontiers of Software Technology, S. 108-122, Springer, Boston.
- [Grün03] Grünbacher, P.: EasyWinWin: Eine groupware-unterstützte Methode zur Erhebung und Verhandlung von Anforderungen. In: Softwaretechnik-Trends der Gesellschaft für Informatik, 23, 2003.
- [JoSW05] John, M.; Jugel, M.; Schmidt, S.; Wloka, J.: Wikis in der Softwareentwicklung helfen. In: Java Magazin, 7, 2005, S. 88-91.
- [KaRy97] Karlsson, J.; Ryan, K.: A Cost-Value Approach for Prioritizing Requirements. In: IEEE Software, 14, 5, 1997, S. 67-74.
- [Lour06] Louridas, P.: Using Wikis in Software Development. In: IEEE Software, 23, 2, 2006, S. 88-91.

- [MaWY06] Majchrzak, A.; Wagner, C.; Yates, D. 2006. Corporate wiki users: results of a survey. In: Proceedings of the 2006 International Symposium on Wikis, Odense, Denmark, ACM Press, New York, NY, 2006, S. 99-104.
- [Onto06] Ontoworld.org: Semantic Wikis. URL [http://ontoworld.org/wiki/Semantic\\_wiki](http://ontoworld.org/wiki/Semantic_wiki) (16.09.2006), 2006.
- [RaJa01] Ramesh, B.; Jarke, M.: Toward Reference Models for Requirements Traceability. In: IEEE Transactions on Software Engineering, IEEE Press, 2001, 27, S. 58-93.
- [RuEP03] Ruhe, G.; Eberlein, A.; Pfahl, D.: Trade-Off Analysis For Requirements Selection. In: Int. Journal of Software Engineering and Knowledge Engineering, 13, 2003, S. 345-366.
- [SaVa01] Saaty, T.L.; Vargas, L.G.: Models, methods, concepts & applications of the analytic hierarchy process. Kluwer, 2001.
- [Somm04] Sommerville, I.: Software Engineering. Addison-Wesley, 2004.
- [TWIK06] TWiki.org: TWiki Success Stories. URL: <http://twiki.org/cgi-bin/view/Main/TWikiSuccessStories> (26.09.2006), 2006.
- [VöKr06] Völkel, M.; Krötzsch M.; Vrandečić D.; Haller, H.; Studer, R.: Semantic Wikipedia. In: Proc. of the 15th Int. Conf. on World Wide Web, Edinburgh, Scotland, May 23-26, 2006.
- [W3C04] W3C 2004: The Web Ontology Language (OWL) Specification. URL: <http://www.w3.org/TR/owl-features/> (16.09.2006), 2004.