



Fraunhofer Institut
Experimentelles
Software Engineering

RISE – Reuse im Software Engineering

Evaluationsmethodik der Kontextermittlung



Autoren:

Jörg Rech
Eric Ras
Björn Decker
Christian Höcht
Lars Kilian

RISE Project, Gefördert durch das
BMBF im Programm „SE 2006“

IESE-Report Nr. 032.05/D
Version 1.2
01. August 2005

Eine Publikation des Fraunhofer IESE

Das Fraunhofer IESE ist ein Institut der Fraunhofer-Gesellschaft. Das Institut transferiert innovative Software-Entwicklungstechniken, -Methoden und -Werkzeuge in die industrielle Praxis. Es hilft Unternehmen, bedarfsgerechte Software-Kompetenzen aufzubauen und eine wettbewerbsfähige Marktposition zu erlangen.

Das Fraunhofer IESE steht unter der Leitung von
Prof. Dr. Dieter Rombach (geschäftsführend)
Prof. Dr. Peter Liggesmeyer
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

Das Projekt RISE beschäftigt sich mit der Wiederverwendung (Reuse) im Bereich Software Engineering.

Dieser Report dient dazu, die Methodik zu beschreiben mit welcher der Kontext (d.h. der Ist-Zustand) bei den Anwendungspartnern ermittelt wird. Dazu wird in diesem Report die Planung sowie Zielsetzung beschrieben und Unterlagen zur Evaluation der Infrastruktur in Software-Unternehmen geliefert. Ergebnisse der durchgeführten Kontextermittlung bei den beiden Anwendungspartnern im RISE Projekt – brainbot technologies AG und empolis – werden in einem weiteren Report veröffentlicht. Ebenso wird eine Verschriftlichung der Interviews und Gruppendiskussionen bei den Anwendungspartnern in einem eigenständigen und vorläufig geheimen Report veröffentlicht.

Schlagworte: Evaluation, RISE, Software Reuse, Software Wiederverwendung, Empirisches Software Engineering

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziele der Kontextermittlung	2
1.2	Begriffsbestimmung und Einordnung	3
2	Ziele der Kontextermittlung	4
2.1	Ermittlung von Verbesserungszielen	4
2.2	Informationen über zu messende Objekte	5
2.2.1	Erfassung der Software Entwicklungs-Abläufe	6
2.2.2	Erfassung der bestehenden Wissenskultur und EM-Abläufe	8
2.2.3	Charakterisierung vorhandener Artefakte (Inhaltstypen)	8
2.3	Variationsfaktoren	9
2.3.1	Organisations-spezifische Variationsfaktoren	10
2.3.2	Projekt-spezifische Variationsfaktoren	10
3	Methoden zur Kontextermittlung	12
3.1	Gruppendiskussion	12
3.1.1	Durchführung von Gruppendiskussionen für RISE	13
3.2	Einzelinterviews	16
3.2.1	Das Tiefeninterview	16
3.2.2	Das Leitfadengespräch	16
3.2.3	Das Fragebogen-Interview	16
3.3	Gruppeninterviews	17
3.4	Artefaktanalysen	17
3.5	Gewährleistung der Gültigkeit der qualitativen Ergebnisse	18
4	Vorgehensweise & Leitfäden zur Kontextermittlung	20
4.1	Phase 0: Problemerkennung	22
4.1.1	Toolwatching: Observation bisheriger Wissensspeicher (Schritt 0a)	22
4.1.2	Interviewleitfaden: Allgemeine Problemerkennung bei der Benutzung von Wikisystemen (Schritt 0b)	23
4.2	Phase 1: Kontextermittlung	24
4.2.1	Interviewleitfaden: Erfassung Wissenskultur/Wissensaustausch (Schritt 2a)	26
4.2.2	Interviewleitfaden: Charakterisierung vorhandener Artefakte (Schritt 2c)	27
4.2.3	Interviewleitfaden: Charakterisierung der WM-Tool-Infrastruktur (Schritt 3)	28

5	Ergebnisse zur Kontextermittlung	29
5.1	Probleme bei der Wikinutzung (Schritt 0b)	29
5.1.1	Gruppendiskussion bei der brainbot technologies AG	29
5.1.2	Gruppendiskussion bei empolis	30
5.2	Erfassung des Software Entwicklungsablaufs (Schritt 1a)	32
5.2.1	Gruppeninterviews bei empolis	32
5.2.2	Interviews bei der brainbot technologies AG	39
5.3	Erfassung der Wissenskultur (Schritt 2a und 2c)	44
5.3.1	Untersuchung bei der brainbot technologies AG	44
5.3.2	Untersuchung bei Empolis	48
5.4	Technische Infrastruktur	49
5.4.1	Befragung bei Empolis	49
5.4.2	Befragung bei der brainbot technologies AG	51
6	Problemsammlung	53
6.1	Verteilungsproblem der Information	53
6.2	Problem der Suche und des Findens	54
6.3	Problem der Qualität & Strukturierung	54
6.4	Problem der Wissensrepräsentation	55
7	Erste Lösungsansätze und Forschungsfragen	56
7.1	Handlungsgebiete	56
7.2	Nutzungsszenarien	56
7.3	Lösungsansätze & Handlungsanweisungen	58
7.4	Forschungsfragen	61
	Referenzen	63

1 Einleitung

Zu einer zielgerichteten Entwicklung des RISE-Systems, wurde ein Evaluationskonzept entwickelt, das während der Laufzeit des RISE Projektes angewendet wird. Das Konzept besteht aus vier Phasen (siehe 0). In der **Phase 0** wird die Benutzung der bisherigen Infrastruktur für die Wiederverwendung (bspw. existierende WIKI-Systeme) während der Softwareentwicklung untersucht um ggf. existierende Probleme zu erheben und zu dokumentieren. Die **Phase 1** befasst sich mit einer detaillierten Ermittlung des Kontextes um daraus Verbesserungspotentiale abzuleiten und die Basis für eine Lösung zu ermitteln.

Aus den Ergebnissen der beiden ersten Phasen werden Anforderungen für das zu entwickelnde System abgeleitet. Das Release 1.0 des RISE Systems wird parallel zur Phase 3 der Evaluation entwickelt. Die Phase 2 der Evaluation ermittelt eine Baseline und stellt Hypothesen auf, die in den beiden späteren Delta Evaluation (Phase 3 bzw. Phase 4) untersucht werden. Die Baseline definiert die Ausgangslage der Software Entwicklung bei den beiden Anwendungspartnern brainbot technologies AG und empolis mit Hinblick auf Wiederverwendung.

Die Vorgehensweise und die Ergebnisse der beiden ersten Phasen werden in diesem Bericht beschrieben. Dabei werden die Ziele der Erhebung als auch die angewendeten Verfahren im Detail beschrieben (siehe Kapitel 2 für die Zielbeschreibung und Kapitel 3 für die Methodenbeschreibungen). Kapitel 4 beschreibt die Anwendung der Methoden bei den einzelnen Anwendungspartnern und Kapitel 5 beschreibt die Ergebnisse der beiden ersten Phasen. Kapitel 6 leitet aus den Ergebnissen der derzeit aktuellen Probleme bei der Software Entwicklung mit Hinblick auf Wiederverwendung ab. Das letzte Kapitel beschreibt die Anforderungen an Release 1.0.

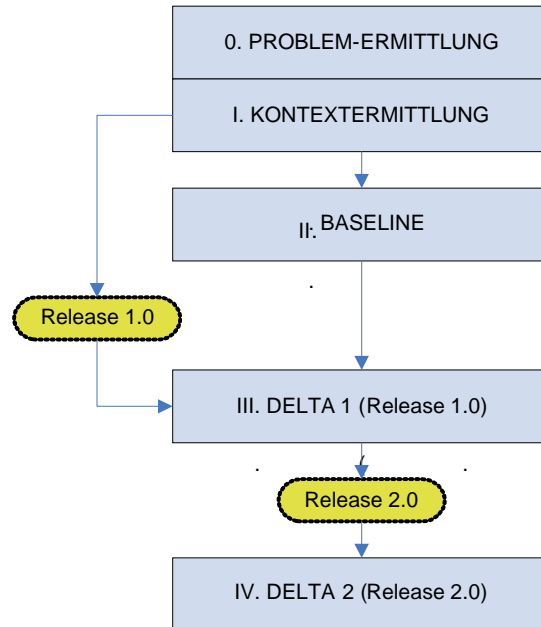


Bild 1 Evaluationsverfahren für RISE

Dieses Dokument richtet sich primär an Personen, die das RISE System in einem neuen Unternehmen einsetzen und die Methodik anpassen wollen und dazu den Kontext in dem Unternehmen ermitteln müssen. Das Einführungsprojekt kann je nach verfügbaren Zeitrahmen, der Verteilung des Unternehmens und der Anzahl der Stakeholder auch von einer einzelnen Person durchgeführt werden. An den Messergebnissen interessierten Personen gibt es einen Einblick in die Vorgehensweise während der Evaluation, in die Datenerhebung und die Auswertung bzw. Synthese von Handlungsanweisungen.

1.1 Ziele der Kontextermittlung

Im ersten Abschnitt des Evaluationsvorhabens des RISE Projektes liegt das Hauptaugenmerk auf der Kontextermittlung. In Diskussion mit Projektpartnern wurde deutlich, dass der Begriff „**Kontext**“ unterschiedliche Verwendung findet. Diese Divergenz unterliegt dem jeweiligen theoretischen Hintergrund, mit dem die einzelnen Projektpartner an den Forschungsgegenstand heran treten. So stellt die Kontextermittlung auch für das FG Pädagogik einen wesentlichen Evaluationsbereiches dar, der zur einer erfolgreichen Entwicklung eines RIKI notwendig ist. Kontext wird dort als die notwendige Umgebung angesehen, in welche eine Informationen in einem Medium eingebunden sein muss, damit sie entsprechend aufgenommen und verarbeitet werden kann. Der hier vorliegende Bericht zielt dagegen auf einen Kontext, der die technischen und methodi-

schen Randbedingungen von Software-Unternehmen (dem Einsatzumfeld des RIKI) in das Zentrum der Betrachtung rückt.

Die Ermittlung des Ist-Zustandes bei den Anwendungspartnern (und später anderen Unternehmen) dient der Ermittlung von Anforderungen für den Aufbau des technischen RISE-Systems das wir RIKI nennen (für Reuse-oriented WIKI). Neben der Erfassung der vorhandenen technischen Infrastruktur (d.h. verwendete Werkzeuge) und der Vorgehensweisen bei der Softwareentwicklung (d.h. gelebte Prozesse und dabei produzierte Dokumente) sollen auch Aspekte wie vorhandene Rollen, der Wissensaustausch und Wissensstrukturen untersucht werden. Für diese Kontextermittlung wurde eine Methodik verwendet, die in diesem Bericht erläutert werden soll.

Die Ergebnisse der Kontextermittlung, die von September 2004 bis Dezember 2004 durchgeführt wurde, werden in einem anderen Bericht verfasst. Basierend auf diesen Informationen wird die Architektur von RISE verfeinert und die Anwendungsbereiche festgelegt. Weitere Evaluation werden sich später mit den Effekten des RISE-Systems auf das Unternehmen befassen.

1.2 Begriffsbestimmung und Einordnung

Im nachfolgenden werden einige Abkürzungen erläutert welche in diesem Report (und im RISE Projekt) Verwendung finden.

RISE: Reuse In Software Engineering

RIKI: Reuse-oriented wIKI (Das technische System das in RISE entwickelt wird)

2 Ziele der Kontextermittlung

Die Ermittlung des Kontextes (Einsatzumfeld) dient der Bestimmung von Randbedingungen die im Unternehmen vorherrschen sowie der Vorbereitung auf die eigentliche Evaluation einer Organisation. Diese Kontextermittlung bildet damit die Grundlage um weitere Arbeiten in RISE und die Architektur von RIKI zu planen. Sie wird in die drei folgenden Kategorien unterteilt:

Ermittlung von (weiteren) Verbesserungszielen: Neben den bereits definierten, projekt-spezifischen Verbesserungszielen werden aktuelle Herausforderungen des Kunden bestimmt. Diese Verbesserungsziele bilden die Grundlage auf denen die Anforderungen für die Definition und Auswahl von Metriken aufgebaut werden.

Ermittlung von Informationen über die Prozesse, Produkte und Werkzeuge: Dies sind in erster Linie Informationen über den Prozess und die damit verbundenen Artefakte und Werkzeuge die bei der Softwareentwicklung eingesetzt werden. Durch Kenntnis des Prozesses kann bestimmt werden, wo die Wiederverwendung optimal eingesetzt werden kann und welche Metriken bei der Auswirkungsermittlung wann erfasst werden. Dabei muss zwischen Informationen über den Prozess (z.B. Aufwand) und damit verknüpfte Artefakte, Rollen und Ressourcen unterschieden werden. Eine Messung muss sich dabei an dem tatsächlich ausgeführten Prozess orientieren, um korrekte Messwerte zu erfassen.

Ermittlung von Informationen über Einfluss- und Normierungsfaktoren: Diese Faktoren beziehen sich auf Eigenschaften der Organisation, in der gemessen wird und damit den jeweiligen Organisationskontext. Als Beispiel sei hier die Mitarbeiteranzahl genannt, die Einfluss auf die Anzahl der neuen Einträge haben kann, und auch als Normierungsfaktor (Anzahl neuer Einträge pro Zeiteinheit pro Mitarbeiter).

Jede dieser Kategorien wird in den Evaluationsleitfäden behandelt. Daran schließt sich eine Zuordnung der jeweils ermittelten Informationen an die eingesetzten Instrumente zur Datenerfassung an.

2.1 Ermittlung von Verbesserungszielen

Neben einer besseren Kenntnis der Anforderungen der Anwendungspartner helfen die Verbesserungsziele, die weitere Charakterisierung und das anschließende Messprogramm zu fokussieren (z.B. die jeweils betrachteten Prozesse). Dies verringert den Aufwand auf Seiten der Anwendungs- und Forschungspartner und steigert so auch die Motivation zur Durchführung des Messprogramms.

Dabei ist eine hohe Überschneidung der Verbesserungsziele bei beiden Anwendungspartnern anzustreben. Durch diese Überschneidung werden ähnliche Objekte zur Messung herangezogen. Weiterhin wird die Vergleichbarkeit der Ergebnisse und so die wissenschaftliche Aussagekraft der Ergebnisse gesteigert.

Um die Verbesserungsziele der teilnehmenden Organisation mit den allgemeinen Verbesserungszielen aus dem RISE-Projekt abzugleichen, müssen sie explizit erfasst werden.

Die grundlegende Fragen dabei sind:

- Was sind die Herausforderungen, und wann und wo halten Sie den Einsatz eines RIKI's für sinnvoll.
- In welchen Abläufen ist externes Wissen oder Wissen aus vorangegangenen Projekten hilfreich (z.B. Requirements-Engineering).

Ein Hilfsmittel zur Unterstützung bei der Definition und Auswahl der Verbesserung, kann die Nutzung der „sieben Fragen“ sein [GQM Handbuch]:

- Was sind die strategischen Ziele der Organisation?
- Was hat einen Einfluss auf die strategischen Ziele der Organisation?
- Wie kann die Performance verbessert werden?
- Was sind die größten Herausforderungen/Probleme?
- Was sind die Verbesserungsziele?
- Wie können die Verbesserungsziele erreicht werden?
- Was sind mögliche Messziele und ihre Prioritäten?

Allerdings kann es sein, dass Verbesserungsziele unklar oder unbewusst sind. In dem Fall sollten prototypische Verbesserungsziele erfasst werden, und nach der Ermittlung der Informationen über zu messende Objekte (in 4.2) erweitert werden.

Die Verbesserungsziele werden im Rahmen von Gruppeninterviews mit Anwendern und Einzelinterviews mit Entscheidungsträgern bei den Anwendungspartnern erfasst (für Details siehe 4.4.)

2.2 Informationen über zu messende Objekte

Grundvoraussetzung für ein erfolgreiches Messprogramm ist eine (ausreichende) Kenntnis über die Objekte, auf welche sich die Messung bezieht. Nur so kann eine valide und möglichst minimal invasive Instrumentierung der Messung erfolgen. Die in diesem Abschnitt beschriebenen Fragestellungen ermitteln und beschreiben diese Objekte. Zusammengefasst wird eine grob-granulare Prozess-Analyse des Software-Entwicklungsprozesses durchgeführt, die insbesondere auf Wissensmanagement-Aspekte eingeht. Als Nebeneffekt dieser Untersu-

chung werden Informationen für den Aufbau der initialen RISE-Ontologie ermittelt.

Zur besseren Übersicht und um die verschiedenen Aspekte klarer herauszuarbeiten, die bei dieser Betrachtung relevant sind, ist dieser Abschnitt in weitere Unterabschnitte unterteilt. Basis für die weitere Ermittlung bildet die Erfassung der Abläufe im Software Engineering. Dieses „Prozessmodell“ wird unter dem Aspekt Wissenskultur / Erfahrungsmanagement durch die Beantwortung weiterer Fragestellungen erweitert. Eine gesonderte Betrachtung der im Software-Engineering und Wissensmanagement relevanten Artefakte dient ebenfalls der Erweiterung, aber auch der Überprüfung der bis dahin erfassten Information.

Die einzelnen Abschnitte sind dabei wie folgt strukturiert: Zuerst wird das allgemeine Ziel erläutert. Danach folgt eine Übersicht über die erwarteten Ergebnisse.

Um die Fokussierung der Ermittlung der Objekte zu steigern und um das Ergebnis überprüfen zu können, sind dabei folgende Anforderungen an die Ergebnisse zu richten:

- Die relevanten Prozesse oder Abläufe sind bekannt. Kriterium dafür ist, ob die zuvor erfassten Verbesserungsziele berührten Abläufe für eine Messung vollständig erfasst sind. Weiterhin sollten weitere Abläufe erfasst werden, die einen Großteil (z.B. insgesamt ca. 70%) des Geschäfts abdecken.
- Anhand der Ergebnisse können Messpunkte gesetzt werden. Insbesondere wird bestimmt, welche vergleichbaren Artefakte bei den Anwendungspartnern gemessen werden soll wollen (z.B. Projekt-Homepage, Anforderungsdokumentation, etc.)
- Die Artefakt-Typen in einem eventuell vorhandenen Wiki und der überwiegende Teil der weiteren zum Management von Wissen Werkzeuge / eingesetzten Infrastruktur-Dienste sind bekannt. Dabei wird auch das aktuelle Wiederverwendungs-Potential dieser Artefakte bestimmt.

2.2.1 Erfassung der Software Entwicklungs-Abläufe

Ziel der in diesem Abschnitts präsentierten Ergebnisse ist es – fokussiert durch die Verbesserungsziele – die Abläufe im Rahmen der Entwicklung der Software zu ermitteln. Neben den Abläufen an sich sollen die dabei verwendeten Artefakte und die nötigen Rollen erfasst werden. Ergebnis ist daher ein grobgranulares „Prozessmodell“, welches existierende Prozesse, Produkte, Werkzeuge und Methoden beschreibt.

Um den Erfassungsaufwand zu minimieren, wird der Einsatz von SE-Referenzprozessen (z.B. Wasserfallmodell, XP) oder Produktmodellen (z.B., logisches Produktmodell) explizit abgefragt.

Grundlegende Vorgehensweise sollte dabei sein, die Angaben durch Beispiel-Artefakte zu verifizieren. Diese Beispielartefakte helfen weiterhin, einen besseren Eindruck über den Prozess zu bekommen .

Um die Fokussierung der Ermittlung der Entwicklungsabläufe zu steigern und um das Ergebnis überprüfen zu können sind dabei folgende Anforderungen an die Ergebnisse zu richten:

- Benötigt wird eine skalierbare Beschreibung der aktuell ausgeführten Abläufe (Prozesse). Initial reicht eine grobgranulare Beschreibung der Prozesse aus. Darauf aufbauend wird - wenn nötig - eine detaillierte Beschreibung der Prozesse erstellt, welche für die späteren Messungen relevant sind.
- Die Übergänge zwischen Prozessschritten / Phasen sollte so klar wie möglich definiert sein.
- Die Artefakte (Produkte & Erfahrungen) sind eindeutig und namentlich bekannt sein und es sollten ca. 1-2 Beispiele pro Artefakt vorliegen.
- Die eingesetzten Werkzeuge (IDE und Infrastruktur) sind bekannt.
- Rollen haben zumindest einen Namen. 1-2 Rollenträger sollten darüber hinaus namentlich bekannt sein. Zusätzlich ist eine klare Aufgaben-, Pflichten- und Rechtebeschreibung wünschenswert.
- Bzgl. der Verteiltheit der Organisation sind die Standorte bekannt. Für die erfassten Prozesse sollte klar sein, welche Schritte an welchen Standorten in welchem Umfang ablaufen.
- Es kann der Grad der Verwendung von agilen SE Verfahren eingeschätzt werden.

Weiterhin werden folgende Informationen für den weiteren Verlauf der Erfassung von Informationsobjekten gewonnen:

- Mögliche Dokumentation zum Selbststudium
- Mögliche Ansprechpartner
- Mögliche Ansatzpunkte zum Toolwatching

Der Bezug dieser Ergebnisse zu anderen Aktivitäten ist wie folgt:

- Das Prozessmodell an sich wird genutzt, um zielorientiert Erfahrungen, Informationen und Wissen abzuschöpfen und an den geeigneten Stellen wiederzuverwenden.
- Die Analyse der existierenden Rollen erlauben es den potentiellen Transfer von Wissen abzuschätzen (Wiederverwendung über Rollen hinweg und über die Zeit (gleiche Rolle)).
- Die anfallenden Artefakte werden genutzt um mögliche Didaktisierungsformalismen zu identifizieren, Verwaltungsmechanismen zu ermitteln oder diese aus existierende Quellen automatisch zu extrahieren.

2.2.2 Erfassung der bestehenden Wissenskultur und EM-Abläufe

Der vorige Abschnitt behandelte die SE-spezifischen Abläufe einer Organisation. In diesem Abschnitt wird beschrieben, wie das so gewonnene Prozessmodell um die Aspekte des Wissens- und Erfahrungs-Management erweitert wird. Genauer sind dies die folgenden Aspekte:

- Erweiterung des Prozessmodells um Aspekte (Unterprozesse / Artefakte) des Wissens- und Erfahrungsaustausches.
- (Weitere) Fokussierung auf die Prozesse, welche im Rahmen der Evaluation betrachtet werden.

Mögliche Ansatzpunkte für eine weitere Erfassung von Informationen zu dem Prozessmodell sind daher:

- Untersuchung von informellen Austauschkanälen (z.B. Raucherzimmer)
- Untersuchung weiterer Tools für eine Inhaltsanalyse (Tool-Watching)
- Wissensbarrieren, deren Überwindung überprüft werden kann (Fragebogen)

2.2.3 Charakterisierung vorhandener Artefakte (Inhaltstypen)

In den vorigen Abschnitten wurden die Artefakte als Bestandteil eines Ablaufs betrachtet. Da der Wiederverwendungsansatz von RISE Artefakt-zentrisch ist, sollten die unterschiedlichen Artefakte und deren Verwendung dediziert betrachtet werden. Dabei wird – und soll – es zu Überschneidungen mit den im vorigen Abschnitt erfassten Informationen kommen. Mittels dieser Überschneidung kann eine Gegenseitige Validierung der Angaben erfolgen.

Ein weiteres Ziel der Charakterisierung vorhandener Artefakte ist es, Informationen zur Erkennung von Artefakten zu sammeln. Daher wird zum einen ein Überblick über verschiedene Artefakttypen erstellt. Zum anderen wird zu jedem

einzelnen Artefakttyp Beispiele und ein initialer Satz von Eigenschaften identifiziert. Basierend auf diesen Angaben können weitere Beispiele eines Typs beim Tool-Watching identifiziert werden und so weitere Trainingsbeispiele erfasst werden. Die Analyse der Inhalte eines eventuell vorhandenen Wiki steht dabei im Vordergrund. Grund dafür ist, dass die von RISE entwickelte Lösung ebenfalls einen Wiki einsetzt, und diese Inhalte daher besonders leicht in diese Lösung zu übertragen sind. Dabei ist eine Befragung am Arbeitsplatz des jeweils Interviewten durchzuführen, um direkt Zugriff auf die jeweilige Information zu erlangen.

Das Ergebnis der Charakterisierung vorhandener Artefakte ist wie folgt:

- Eine Liste von Artefakttypen samt Charakterisierung
- Eine Tabelle, welche die Artefakttypen den jeweils zur Bearbeitung eingesetzten Werkzeugen zuweist.

2.3 Variationsfaktoren

In dem vorigen Kapitel wurden die Objekte identifiziert und charakterisiert, die Objekte der Messungen sind. Die hier vorgestellten Metriken geben eine Charakterisierung des Kontext und damit mögliche Variationsfaktoren an, in dem gemessen wird.

Diese Variationsfaktoren sind nicht die Metriken, an deren Messung man primär interessiert ist. Allerdings fließen sie auf zwei Weisen in diese primären Metriken ein:

- Als Einfluss-Faktor, d.h., in wie weit der jeweilige Kontext-Faktor eine Messung beeinflusst (z.B. hohe Verteiltheit führt zu mehr Wissensaustausch über WIKI)
- Als Normierungsfaktor, d.h. mit der eine von Kontext-Faktoren abhängige Metrik auf eine vergleichbare Basis gebracht wird. (Bsp.: Anzahl Beiträge / Anzahl Mitarbeiter)

Darüber hinaus verschaffen die Variationsfaktoren einen Eindruck über einen Einstieg in die Umstände des jeweiligen Messprogramms.

Die Einfluss und Normierungsfaktoren sind dabei in zwei Gruppen unterteilt: Organisations-spezifische Variationsfaktoren gelten für die gesamte Organisation und geben damit relevante organisatorische Rahmenbedingungen an. Projekt-spezifische Variationsfaktoren geben die Umstände eines Projekts an und sind zum Teil einer Spezialisierung der organisations-spezifischen Variationsfaktoren.

2.3.1 Organisations-spezifische Variationsfaktoren

Die im folgenden vorgestellten Faktoren gelten organisationsweit und müssen daher nur einmal (z.B. von einer Führungskraft) ausgefüllt werden. Diese Angaben dienen auch der Plausibilitätsprüfung von projekt-spezifischen Variationsfaktoren (z.B. durchschnittliche Projektlaufzeit im Vergleich zur Laufzeit des aktuellen Projektes.)

Faktor	Ausprägung	Einfluss
Generelles Lebenszyklusmodell:	XP, Iterativ, Wasserfall, ad hoc	Genauigkeit der Messungen, indirekter Einfluss auf Kommunikation der Mitarbeiter
Verteilung	Anzahl der Standorte	Hohe Verteiltheit: bedingt explizite Kommunikation nötig → Einfluss auf Übertragbarkeit.
Größe (#Mitarbeiter)	Anzahl der Personen (Gesamt / Entwickler)	Normierung über die Anzahl der Personen möglich (z.B. bei Anzahl der Beiträge)
Durchschnittliche Projektlaufzeit?	Angabe eines Intervalls in Personenmonaten	Ungefähre Einschätzung der Projektgröße
Wochenarbeitszeit/ Durchschnittliche Arbeitsdauer	Angabe Stunden pro Woche	Hinweis auf Überstunden → Hinderungsgrund für Nutzung

2.3.2 Projekt-spezifische Variationsfaktoren

Die projekt-spezifischen Variationsfaktoren geben nähere Auskunft über die Eigenschaften des jeweiligen Projektes, in dessen Rahmen die Messung stattfindet.

Faktor	Ausprägung	Einfluss
Typ des Projekts	Infrastrukturentwicklung (Domain Engineering), Anwendungsentwicklung (Application Engineering), Wartung (Kundenaufträge, eigene Produktweiterentwicklung, Wartungsprojekte, ...)	Art und Umfang der generierten und benötigten Informationen
Größe (# Mitarbeiter)	# Mitarbeiter # Mitarbeiter, auf Vollzeitstellen umgerechnet	Grosse Projekte neigen eher zu expliziterer Dokumentation. Ermöglicht weiterhin Normierung von Metriken
Geschätzter Aufwand	Aufwand, Ertrag	Grosse Projekte neigen eher zu expliziterer Dokumentation. Ermöglicht weiterhin Normierung von Metriken
Verteiltheit der Zusammenarbeit	Anzahl der im Projekt beteiligten Filialen	Verteiltheit impliziert höherer Anteil an expliziter Dokumentation

3 Methoden zur Kontextermittlung

Qualitative Methoden werden dann eingesetzt, wenn keine quantitativ ausgerichteten Fragen im Mittelpunkt der Untersuchung stehen, sondern es in erster Linie um qualitative Sachverhalte geht. Im Folgenden werden die qualitativen Methoden, die im Rahmen von der Kontextermittlung in RISE verwendet wurden beschrieben. Dazu gehören:

- Gruppendiskussion
- Einzelinterview
- Gruppeninterview
- Artefaktanalyse

3.1 Gruppendiskussion

Die *Gruppendiskussion* ist ein Verfahren der allgemeinen Meinungs- und Einstellungsforschung. Bei dieser besonderen Art einer qualitativen Befragung diskutiert eine Gruppe von Personen strukturiert von einer Moderatorin bzw. einem Moderator über ein bestimmtes Thema: Eine Methode dieser qualitativen Forschung stellen die so genannten Fokusgruppen dar. Generell gibt es viele verschiedene Definitionen von Gruppendiskussionen, die in der unterschiedlichen Forschungspraxis von Beobachtungen von Gruppenprozessen bis hin zum Abfragen von Gruppeninformationen begründet sind. So reichen die Begriffe von „Gruppenexperiment“ (vgl. Dreher 1982), „focus group“ (vgl. Merton 1987), „Delphi-Methode“ bis hin zum Gruppeninterview. Im Rahmen von RISE verwenden wir den Begriff Gruppendiskussion.

David L. Morgan (1997) bezeichnet die Gruppendiskussion als eine Erhebungsmethode. Lamnek sagt, dass die Daten durch die Interaktionen der Gruppenmitglieder gewonnen werden, wobei die Thematik durch das Interesse des Forschers bestimmt wird (vgl. Lamnek 1998, 27).

Bei der Gruppendiskussion handelt es sich nicht um ein Experiment da gewisse methodische Bedingungen wie: Operationalisierbarkeit, Meßbarkeit, Reproduzierbarkeit, Standardisierbarkeit und Kontrolle der vergleichbaren Gruppen nicht garantiert werden können. Die Gruppendiskussion ist ein hypothesengenerierendes und kein hypothesenprüfendes Verfahren.

Im Gegensatz zu Gruppeninterviews (siehe nächstes Kapitel) steht in der Gruppendiskussion die „Bedeutung von Interaktions-, Diskurs und Gruppenprozessen für die Konstitution von Meinungen, Orientierungs- und Bedeutungsmustern“ (Bohnsack 1999, s. 123) im Mittelpunkt. Bei Gruppeninterviews handelt

es sich um eine rationale Methode der Einzelbefragung. Das Kollektiv wird hier nicht als Gegenstand der Erhebung wahrgenommen. Gruppendiskussionen befassen sich vor allem mit dokumentarischen Methoden und den kollektiven Aspekten der Diskussion.

Grundsätzlich kann man zwischen ermittelnder und vermittelnder Gruppendiskussion unterscheiden. Die vermittelnde Gruppendiskussion dient der Inszenierung von Gruppenprozessen. Ermittlende Gruppendiskussion haben zum Ziel, Informationen und Befunde inhaltlicher Art oder über gruppendynamische Prozesse zu erheben, RISE verwendet ermittelnde Gruppendiskussionen.

Die Methode der Gruppendiskussion bietet folgende Vorteile:

- höhere Realitätsnähe bei der Meinungsäußerung
- wechselseitige Stimulation im Meinungs austausch
- Informationen für Forscher; Lernprozess für Betroffene weil z.B. noch unbekannte Probleme entdeckt werden
- Die Offenheit insbesondere gegenüber der Untersuchungssituation und den beteiligten Personen ist bei der Gruppendiskussion mehr als beispielsweise bei Interviews gegeben, da die Teilnehmer den Verlauf und die Themenhierarchie durch in großem Maße selbst bestimmen können
- multilaterale Interaktion im Gegensatz zur bilateralen Interaktion im konventionellen Interview
- rasche, effiziente und offene Methode, zudem ist sie meist auch deutlich preisgünstiger als quantitative Erhebungen oder Einzelinterviews

3.1.1 Durchführung von Gruppendiskussionen für RISE

Auswahl der Teilnehmer und Diskussionsplanung

Für die Kontextermittlung bei RISE wird versucht, Teilnehmer mit unterschiedlichen Rollen in der Software Entwicklung, auszuwählen. Die optimale Größe einer Gruppe liegt bei 5 Teilnehmern. Wichtige Diskussionsregeln werden vorab vom Diskussionsleiter festgelegt und ggf. abgestimmt, falls mehrere Diskussionsleiter an der Gruppendiskussion teilnehmen. Es findet eine Vorab-Thematisierung und eine Vorab-Fixierung individueller Standpunkte zwischen den Diskussionsleitern statt. Die Vorab-Thematisierung dient vor allem dazu, am Anfang der Gruppendiskussion den Teilnehmer einen groben Rahmen vorzugeben, indem man das Leitthema der Diskussion vermittelt. Einen Fragenkatalog für die Nachfragephase wird von den Diskussionsleitern erstellt, der nach der freien Diskussionsphase verwendet werden kann. So können für die Forscher relevante Themen noch mal aufgegriffen und mit den Teilnehmern bearbeitet werden.

Durchführung der eigentlichen Diskussion:

Die eigentliche Durchführung der Diskussion besteht aus drei Phasen: Einführung, eigentliche Gruppendiskussion und Nachfragephase.

Während der Einführungsphase gibt der Diskussionsleiter Vorab-Informationen. Die Einführung beginnt mit der Vorstellung des Diskussionsleiters. Der Diskussionsleiter bespricht mit den Teilnehmern die nötigen Formalien: dazu gehören z.B. Absprachen über die Aufzeichnung (evtl. Tonbandaufnahme), Verwertung der Ergebnisse, grundsätzlicher Ablauf der Diskussion und Dauer. Die Beteiligten werden bereits vor der Diskussion über die Ergebnissrückkoppelung und die Möglichkeit von Korrekturen und Ergänzungen informiert. Hierdurch kann auch eine höhere Offenheit im Interview erreicht werden, da die Teilnehmer die Möglichkeit haben, vor der weiteren Verwendung ihrer Aussagen diese nach richtiger Rekonstruktion und Korrektheit zu überprüfen. Der wichtigste Teil der Einführung ist die Präsentation des Grundreizes. Der Diskussionsleiter schildert kurz den Sinn und Zweck der Gruppendiskussion und versucht die vorher abgestimmten Themen durch einen Grundreiz einzuleiten. Dabei soll vermeiden werden, dass die Diskussionsthemen explizit genannt werden, um den Diskussionsverlauf von vornherein nicht zuviel einzuschränken. Für die einzelnen durchgeführten Gruppendiskussionen bei RISE werden die Themen in den späteren Kapiteln angegeben.

Der Diskussionsleiter hält sich während der eigentlichen Diskussionsphase zurück und überlässt den Teilnehmern weitestgehend die Bestimmung des Verlaufs und der Themenhierarchie. Er sollte nur eingreifen wenn er den Eindruck hat, dass die Diskussion unter den Teilnehmern stockt oder droht zu stoppen, bzw. die Diskussionsinhalte sehr stark von den eigentlichen Themen abweichen, aber auch zum Schlichten von Meinungsverschiedenheiten. Durch provokante Statements kann der Diskussionsleiter die Diskussion wieder anregen oder zum gewollten Thema zurückführen. Durch gezieltes Nachfragen können bestimmte Aussagen noch mal tiefgründiger besprochen und geklärt werden. Durch Paraphrasierungen kann der Diskussionsleiter prüfen ob er den Sachverhalt richtig erfasst hat. Auch durch das Aufzeigen von Konsequenzen kann der Diskussionsleiter z.B. weitere Themen einleiten und neue Anreize geben. Bei den RISE Gruppendiskussionen nahmen immer mehrere Diskussionsleiter teil. So konnten unterschiedliche Schwerpunkte abgedeckt werden (siehe folgende Kapitel für Details). Alle Diskussionsinhalte wurden auf Tonband aufgenommen und die Diskussionsleiter protokollierten die wesentlichen Aussagen der Teilnehmer mit. Die Dauer dieser Phase der Diskussion dauerte meistens eine bis anderthalb Stunden. Dann wurde ein ca. 10-15 minütige Pause einberufen. Während dieser Pause berieten sich die Diskussionsleiter und wählten einzelne Fragen aus dem Nachfragekatalog aus, die in der Nachfragephase nochmals explizit gestellt werden sollten. Meistens wurden jedoch schon viele Fragen aus dem Katalog schon während der Gruppendiskussion ausführlich besprochen, sodass nur wenige Fragen gestellt werden mussten.

Durch diese Fragen sollen bestimmte Themen die während der Diskussion noch nicht ausführlich genug besprochen wurden, noch mal vertieft werden. Die Dauer der letzten Phase betrug oft nur 15 Minuten.

Auswertung der Gruppendiskussion

Neben den Tonbandaufnahmen liegen auch die Protokolle der einzelnen Diskussionsleitern vor. Zunächst muss das aufgezeichnete Interview in eine Textform überführt werden. Bei wissenschaftlichen Untersuchungen wird hierzu üblicherweise eine vollständige Transkription des Interviews durchgeführt (vgl. König/Volmer, 1994, S. 121; Lamnek, 1995, S. 108).

Der aufzubringende Aufwand einer Transkription ist extrem hoch. Aus diesem Grund wurde auf eine vollständige Transkription aller Interviews verzichtet. Alternativ wurden ausgewählte inhaltsrelevante Teile paraphrasiert und transkribiert, wobei hier der Nachteil entsteht, dass der Interviewverlauf, also die Gesprächsdynamik, Aussagenreihenfolge, etc., nicht mehr oder schwierig nachvollziehbar ist. Wenn das RISE-Team den Eindruck hatte, dass die Aussagenreihenfolge nebensächlich war oder die Gesprächsdynamik keine große Rolle bei der Analyse spielte, so wurden die oft sehr detaillierten Protokolle der Diskussionsleiter als Grundlage für die Auswertung genutzt. Sollte sich doch noch herausstellen, dass bestimmte Teile der Protokolle nicht detailliert genug waren so wurde die entsprechenden Teile der Tonbandaufnahmen zusätzlich verwendet. Dies war unter anderem auch der Grund warum immer mehrere Diskussions Teilnehmer gleichzeitig an den Gruppendiskussionen teilnahmen. So wurden die Diskussionen aus unterschiedlichen Gesichtspunkten mitprotokolliert. Für die einzelnen Gruppendiskussionen wird angegeben welches Auswertungsverfahren wir in den einzelnen Phasen angewendet haben.

Vor der Transkription wurden Regeln des Umgangs mit nonverbalen Aspekten (z.B. Pausen, Lachen, Unterbrechungen) festgelegt. Das Transkript sollte schließlich von den Diskussionsleitern hinsichtlich Tipp- und Hörfehlern, Unstimmigkeiten, etc. überprüft und korrigiert werden.

In einem weiteren Arbeitsschritt wird das Transkript um redundante Aussagen, inhaltsfreie Füllwörter, etc. gekürzt. Anschließend erfolgt eine Paraphrasierung der einzelnen Aussagen, um die inhaltlichen Kernaussagen zu eruieren und den Textumfang weiter zu reduzieren. Bei der Reduktion soll eine erste Interpretation der Aussagen vermieden werden; es sollten lediglich einzelne Sätze in Form von Kernaussagen zusammengefasst werden. Nun wird das Textmaterial sortiert und strukturiert, um die Textaussagen auf die zugrunde liegenden Forschungsfragen beziehen zu können. Hierzu können spezielle Textanalyseprogramme wie z.B. *MaxQDA* oder *Atlas.ti* eingesetzt werden. Eine Zuordnung findet in erster Linie anhand der festgelegten Themen statt. Mehrfachzuordnungen möglich sind, wenn eine Aussage sich auf mehrere Themen bezieht. Nachdem alle Aussagen auf diese Weise zugeordnet und sortiert worden sind, erfolgt im weiteren Auswertungsprozess eine Strukturierung der Aussagen mit

Hilfe eines Kategoriensystems. Ein Kategoriensystem umfasst mehrere Kategorien, die auf verschiedenen hierarchischen Ebenen (z.B. gegliedert nach Konkretheits- bzw. Abstraktionsniveau) angeordnet und vernetzt sein können.

3.2 Einzelinterviews

Beim Einzelinterview gibt es im qualitativen Verfahren drei Methoden:

3.2.1 Das Tiefeninterview

Hier wird der Interviewer seine Fragen nicht nach vorgegebenen Fragen gestalten, sondern sein Gespräch strukturiert sich individuell durch die Antworten des Gesprächspartners. Dadurch ergibt sich für den Interviewer die Möglichkeit, sehr in die Tiefe zu gehen und einzelne Themen intensiv auszuforschen, ohne an ein festes Schema gebunden zu sein.

Es ist klar, dass das Ergebnis solcher Interviews in erhöhtem Maße subjektiv durch den Interviewer gefärbt wird, und dass auch die Geschicklichkeit des Interviewers direkten Einfluss auf die Verwertbarkeit der Antworten hat.

Darüber hinaus ist natürlich die allgemeine Gültigkeit solcher Ergebnisse nicht gewährleistet.

3.2.2 Das Leitfadengespräch

Auch das Leitfadengespräch hat keinen Fragebogen mit festen Themen als Grundlage, sondern nur vorgegebene Themen. Dabei haben die InterviewerInnen die Freiheit, die Fragen je nach Reaktion des Gegenübers selbst zu formulieren, um Widerstände in den Antworten beim Interviewten zu überwinden. Sie protokollieren den Gesprächsverlauf stichwortartig.

Auch hierbei ist der subjektive Einfluss der InterviewerInnen nicht zu vernachlässigen. Die Auswertbarkeit des Leitfadengesprächs ist etwas einfacher als die des Tiefeninterviews.

3.2.3 Das Fragebogen-Interview

Die Fragen sind fest vorgegeben, meistens als "multiple choice"-Befragung. Dadurch wird die Auswertung stark vereinfacht, weil sie problemlos mengenanalytisch über einen Rechner durchgeführt werden kann. Individuelle Meinungen und Zwischentöne gehen dabei allerdings unter, das Ergebnis wird in gewisser Weise vergrößert.

Auch ist das Risiko nicht zu unterschätzen, dass durch die Art der vorgegebenen "multiple choice"-Antworten das Ergebnis ganz erheblich beeinflusst werden kann.

3.3 Gruppeninterviews

Unter der Methode *Gruppeninterview* verstehen wir eine Gruppenbefragung. Es handelt sich dabei um eine zeitökonomische Variante der Einzelbefragung. Der Interviewer stellt die gleichen Frage an alle zu interviewenden Personen gleichzeitig. Im Gegensatz zur Gruppendiskussion ist die Situation der Gruppe kein Gegenstand der Analyse. Eine detaillierte Beschreibung der Interviewmethoden finden sie im vorgehenden Kapitel.

3.4 Artefaktanalysen

Da die Nutzbarkeit der bereits eingeführten WIKI-Systeme inzwischen an ihre Grenzen stößt und eine Strukturierung zwingend notwendig erscheint, soll zunächst mit Hilfe von Artefaktanalysen die individuelle Form der Strukturierung von Inhalten untersucht werden. Artefakte bilden einen zentralen Bestandteil der organisationalen Lebenswelt und stellen ein wertvolles Material für das Verständnis der betrieblichen Kommunikation dar (vgl. Froschauer 2002, S. 362).

Zu diesem Zweck wurden Mitarbeiter der Anwendungspartner Empolis und der brainbot technologies AG gebeten, Screenshots ihrer Ablagestrukturen anzufertigen. Gleichzeitig wurden die Mitarbeiter am Arbeitsplatz befragt, um ihre individuellen Ablagestrukturen (E-Mail, Dateiablage, Browser-Lesezeichen) kommentieren zu können.

Ziel der Untersuchung war es, individuelle Strukturierungsformen zu identifizieren. Hierzu wurden folgende Leitfragen formuliert:

- In welcher Tiefe bzw. Breite werden Strukturen angelegt?
- Orientieren sich die individuell angelegten Strukturen an bestehenden Strukturen?
- Gibt es Strukturen, die brach liegen?
- Lassen sich anwenderspezifische Strukturierungsformen erkennen?

Mit Hilfe der Artefaktanalyse sollen Hypothesen bzw. zunächst Forschungsfragen generiert werden, wie Ablagestrukturen gestaltet werden müssen, damit sie individuellen Nutzungsanforderungen Rechnung tragen:

- Wie kann der Zugriff sowohl auf aktuelle als auch auf relevante Informationen erleichtert werden?

- Anhand welcher Kriterien wird entschieden, dass Informationen nicht mehr „aktuell“, jedoch weiterhin relevant sind?
- Wie lässt sich verhindern, dass Inhalte brach liegen und nur mit Mühe auffindbar sind?
- Wie können nutzerspezifische Anforderungen durch eine gezielte Strukturierung bzw. Darstellung von Inhalten erfüllt werden?

Das zu entwickelnde prototypische Anwendungssystem soll den Mitarbeitern einen fokussierten Blick auf die für sie gerade benötigte Information ermöglichen (Wissensmanagement) und erleichtern (Usability).

Außerdem galt den in den Unternehmen eingeführten WIKI-Systemen zur kollaborativen Wissenserzeugung das Interesse der Untersuchung. Hierzu wurden (vorwiegend im Unternehmen Empolis) die bestehenden Inhalte im Intranet einer ersten Artefaktanalyse unterzogen.

Artefaktanalyse sind anderen Untersuchungsmethoden häufig dann überlegen, wenn um die Erforschung von Materialformen geht, bei denen sich keine Sequenzialität bzw. feste Strukturierung erkennen lässt. Die zugrunde liegende Struktur ist in den untersuchten Materialien zwar grundsätzlich enthalten, aber nur durch Rekonstruktion erschließbar, durch:

- den Prozess ihrer Vergegenständlichung (Produktionskontext)
- ihr Auftreten (Alter, Reihenfolge, Wiederholungen etc.)
- ihren Gebrauch (Konsum, Konservierung)
- ihre organisationalen Beziehungen (soziale Bedeutung)

Artefaktanalysen sind mit einem hohen Aufwand verbunden, da die vielfältigen Materialien von den beteiligten Forschern erschlossen werden müssen, um einen Zugang zu Organisationen zu bekommen. Die relativ leichte Zugänglichkeit von Artefakten macht sie jedoch für den Forschungseinstieg zu einer unverzichtbaren Orientierungshilfe für die Planung der weiteren Forschungsarbeit und die Vorbereitung anderer Erhebungsverfahren wie etwa Einzel- oder Gruppeninterviews .

3.5 Gewährleistung der Gültigkeit der qualitativen Ergebnisse

In einer Interviewsituation, in der auf dem Wege der direkten Kommunikation Informationen erhalten werden, spielen individuelle und wechselseitige Interpretationen von Aussagen eine große Rolle. Ob ein auf dieser Situation basierendes Ergebnis letztlich Gültigkeit besitzt, kann nur dialogisch sinnvoll erfolgen. Insofern wird die Gültigkeit der zu erwartenden Ergebnisse im Rahmen von RISE zunächst zwischen Diskussionsleitern und den Beteiligten kommunika-

tiv ausgehandelt indem die Ergebnisse den Teilnehmern vorgelegt werden. So kann heraus gefunden werden, ob Zusammenfassung, Strukturierung und Interpretation der Diskussionsaussagen in Einklang mit der Sichtweise der Teilnehmer steht oder ob Missverständnisse oder Fehlinterpretationen vorliegen.

4 Vorgehensweise & Leitfäden zur Kontextermittlung

Zur Trennung von Ziel, Technik und Ergebnis werden in diesem Abschnitt die Leitfäden für die Kontextermittlung aufgeführt. Die hier vorgestellten Leitfäden dienen dazu mit einem (neuem) Kunden die Kontextermittlung durchzuführen.

Wie in 0 dargestellt wird eine Problemerkennung (Phase 0) und Kontextermittlung (Phase 1) durchgeführt. Die Problemerkennung dient der vorläufigen groben Identifikation von Problemen und Herausforderungen mit den existierenden Wissensspeichern (z.B. im Falle von empolis ein WIKI-System namens MASE) sowie deren Vor- und Nachteile. Dabei wird mittels einer Gruppendiskussion im Schritt „Problemerkennung“ der zwischenmenschlichen Austausch von Informationen in einem typischen Projektteam beleuchtet. Der Schritt „Untersuchung der Wissensspeicher“ verwendet die Methode „Artefaktanalyse“ um basierend auf existierenden Informationssystemen ein Gefühl für wiederverwendungswürdige Informationen zu erhalten.

Die Kontextermittlung hingegen beschäftigt sich dann mit der systematischen Ermittlung der Prozesse, Produkte und technischen Infrastruktur um konkrete Probleme identifizieren und herauszuarbeiten sowie erste Lösungsansätze zu entwerfen. Die drei Teilphasen beschäftigen sich dabei mit den Entwicklungsprozessen, der Wissenskultur sowie der technischen Infrastruktur in die eine spätere Lösung integriert werden soll. Dabei verwendet man im Schritt „Erfassung des Software-Entwicklungsablaufes“ ein Gruppeninterview mit mehreren Produkt- und Entwicklungsmanagern um einen typischen Entwicklungsprozess zu erarbeiten. Der Schritt „Erfassung der Wissenskultur“ wird mittels einer Gruppendiskussion geführt um den typischen Wissensmanagementprozess zu ermitteln. Abschließend führen Einzelinterviews oder gezielten Fragebögen auf die technische Infrastruktur für den Schritt „Erfassung der technischen Infrastruktur“.

Für die Durchführung der Kontextermittlung wurde ein Zeitraum von ca. 3 Monaten eingeplant (Anfang September bis Ende November 2004). Basierend auf unterschiedliche Urlaubsplanungen, Sommer- Herbst- und Weihnachtsferien sowie anderen Projekten sollte man bei einer solchen Kontextermittlung auch mehr Zeit einplanen.

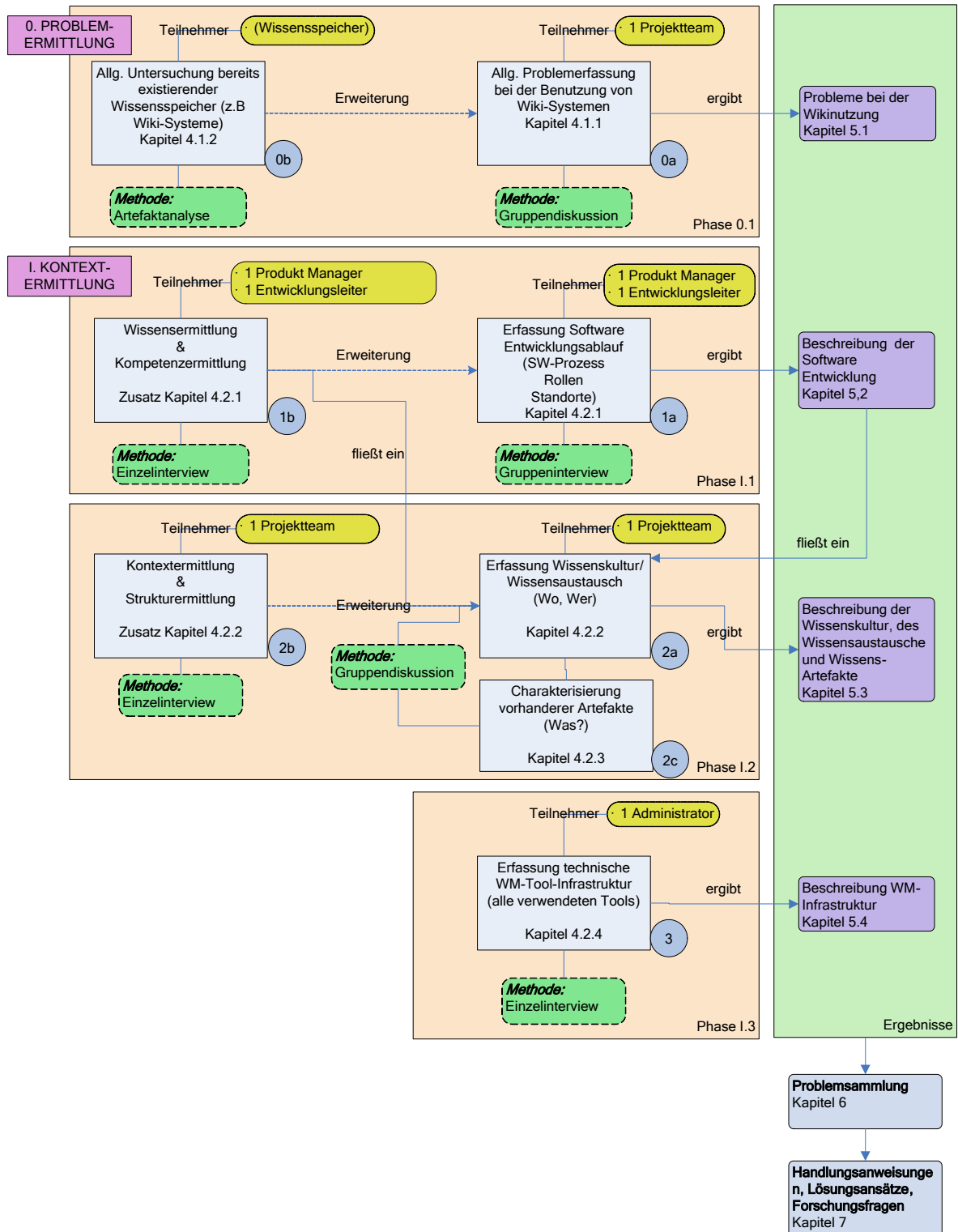


Bild 2 Übersicht über die Schritte der Kontextermittlung

4.1 Phase 0: Problemerkennung

Zur Ermittlung von Problemen mit existierenden Wissensspeichern, die zur Wiederverwendung eingesetzt werden könnten, werden in Phase 0 verschiedene Einflussfaktoren ermittelt. Die Methoden, welche bei der Problemerkennung verwendet wurden, sind in den nachfolgenden Abschnitten ausführlich beschrieben.

4.1.1 Toolwatching: Observation bisheriger Wissensspeicher (Schritt 0a)

Die erste als „Toolwatching“ benannte Methode wird zur Bestimmung der Wissensquellen und dem Umgang mit den darin enthaltenen Informationen verwendet.

Dabei werden die Wissensquellen in Form von WIKIs, Dateisystemen, Intranets, etc. von externen Personen (z.B. den Wissenschaftlern) untersucht. Auf folgende Merkmale ist dabei zu achten:

- Welche *Strukturen* werden zur Navigation und Kategorisierung verwendet? Dabei ist zwischen der Navigations- und Seiten-Struktur zu unterscheiden (Bsp: Die Verzeichnisse in einem Dateisystem und die Überschriften in einer Datei stellen beides Strukturierungsmerkmale dar)!
- Gibt es Gemeinsamkeiten bei den Strukturen verschiedener Wissensspeicher?
- Welche *Typen* von Informationen wurden bisher in den Wissensspeichern abgelegt? (Beispiel für Typen sind: Erfahrungen, Beobachtungen, Problemlösungspaare, Beschreibungen von Design Patterns oder Technologien, ...)
- Wer waren die *Produzenten* dieser Wissensartefakte?
- Wer waren die (potentiellen) *Konsumenten* dieser Wissensartefakte?
- Wie sehen die bisherigen *Zugriffsszenarien* aus? Wer soll produzieren und wer konsumieren? Wer kümmert sich um die Navigationsstruktur (oder Ontologie)? Gibt es Templates für spezifische Typen (z.B. Anforderungen) von Wissensinhalten?
- Wie sehen die bisherigen *Zugriffsmechanismen* aus? Gibt es eine Navigation? Eine Suchfunktion?
- In welcher *Form* werden Informationen in den Wissensartefakte (z.B. Datei oder WIKI-Seite) kodiert? Beinhaltet ein Wissensartefakt viele oder wenige Kontext-Informationen die beim Verständnis des Inhaltes hilfreich sind?
- Gibt es Gemeinsamkeiten bei Wissensartefakten unterschiedlicher Wissensspeicher?

- Lassen sich Unterschiede feststellen zwischen der *individuellen Art*, neue Strukturen anzulegen und bestehende Strukturen zu nutzen?

4.1.2 Interviewleitfaden: Allgemeine Problemerkfassung bei der Benutzung von Wikisystemen (Schritt 0b)

Vier Mitarbeiter werden problemzentriert zu möglichen Ursachen für das Scheitern des bereits eingeführten WIKI-Systems befragt. Die Gruppendiskussion soll eine angenehme Atmosphäre ermöglichen, die es allen Beteiligten erlaubt, den Grad der Aktivität bzw. Passivität selbst zu bestimmen.

Mit Hilfe der Gruppendiskussion sollen auf ökonomische Weise die verschiedenen Positionen der Mitarbeiter ermittelt werden, um daraus mögliche Ursachen für das Scheitern von WIKI-Systemen ableiten zu können. Im Verlauf der Gruppendiskussion soll den besonderen Bedingungen kleiner und mittlerer Unternehmen Rechnung getragen werden.

An der Gruppendiskussion nehmen folgende Mitarbeiter teil: Geschäftsführer, Chefentwickler, Projektmanager, Marketing-Verantwortlicher. Die Gruppendiskussion beginnt mit einem erzählgenerierenden Ausgangsstimulus, der den thematischen Rahmen absteckt und der Gruppe ausreichend Freiraum zur eigenen Ausdifferenzierung des thematisch abgesteckten Rahmens erlaubt. Die Gruppe soll in dieser Phase möglichst selbstständig und ohne ständiges Nachfragen durch die Interviewer diskutieren. Erst wenn die Interviewer zu der Ansicht kommen, dass das „immanente Potential“ der Gruppe erschöpft ist, beginnt eine Nachfragephase. In dieser Phase werden untersuchungsrelevante Aspekte angesprochen, insofern sie noch nicht ausreichend durch die Gruppe angesprochen wurden. Im Verlauf der Sitzung wird von mindestens einem der Interviewer ein Protokoll angefertigt, welches ggf. am Ende durch gezielte Nachfragen an die Gruppe ergänzt wird.

Was glaubt ihr, könnten die Gründe sein, warum das WIKI bei euch wieder in der Versenkung verschwunden ist?

Dauer: ca. 120 Minuten

Material: Whiteboard

Methode: Gruppendiskussion

- Was war die Motivation, ein WIKI einzuführen? (Entstehungshintergrund)
- Welche Inhalte im WIKI waren besonders nützlich? (Inhalte)
- Wie wurden die Inhalte im WIKI strukturiert? (Struktur)
- Auf welche Art und Weise tauscht ihr euer Wissen ohne WIKI aus? (Wissensaustausch)
- Welche Informationen ließen sich im WIKI nur mit Mühe auffinden? (Nutzbarkeit, Auffindbarkeit)

- Zu Gunsten welcher (Informations-) Angebote wurde die Aktivität im WIKI eingestellt? (Infrastruktur)
- Welche (Entwicklungs-) Prozesse wurden durch das WIKI nicht ausreichend unterstützt? (SE Aufgaben und –Prozesse)
- Wie müsste ein WIKI gestaltet sein, um euch bei der Arbeit zu unterstützen? (Verbesserung)

4.2 Phase 1: Kontextermittlung

Während der Phase 1 wurde verschiedenen Methoden angewendet. Sie werden für die einzelnen Schritte (wie in Bild 0 dargestellt) angegeben.

Skizzieren sie bitte kurz die Phasen eines typischen Projektes

Dauer: ca. 120 Minuten Material: Whiteboard Methode: Gruppeninterview

Ziel: Diese Frage dient dazu, einen reinen Überblick über die Abläufe der Organisation zu erhalten. Ebenfalls eine Gewichtung der Prozesse um später gezielt und mit größtmöglichem Einfluss Verbesserungen einzubringen.

Anmoderation: Der betrachtete Ausschnitt umfasst dabei den Projektstart bis zum Projektabschluss. Wenn es kein typisches Projekt gibt, nehmen sie bitte einen Projekttyp, der zu den am häufigsten durchgeführten Typen gehört.

Weitere Fragen zu den Projektabläufen:

- Gibt es eine Dokumentation zu diesen Prozessen? (Prozessmodelle um sich evtl. weiter einzulesen?)
- Woran sind die Übergänge zwischen diesen Phasen erkennbar? (Welche Trigger gibt es oder an welchen Merkmalen kann man den Prozessübergang erkennen?)

Zur gezielten Unterstützung der Nutzer von RISE müssen wir Produkte ermitteln mit denen die Benutzer in Berührung kommen und aus denen Informationen zur besseren Benutzung von RISE extrahiert werden können.

- Welche Dokumente / Informationen werden in diesen Phasen erzeugt oder verwendet?
- Mit welchen Tools werden diese Dokumente / Informationen verwaltet?
- Welche Rollen sind an diesen Phasen beteiligt? Nennen Sie bitte einen oder zwei Vertreter dieser Rolle. Welches Rollenmodell existiert im Unternehmen? z.B. Managementrollen (Projektmanager, Qualitätsmanager), Entwicklungs-

rollen (Entwickler, Tester), Administrationsrollen (EF-Manager, Ontologiewar-ter), Supportrollen (Sekretärin)

- Wie sieht die Zusammenarbeit und der Austausch mit anderen Projekten / anderen Standorten im Rahmen der Prozesse aus? Werden Subsystemen von mehrere Mitarbeiter bearbeitet die sich nicht direkt kennen oder treffen können (z.B. wegen Standort)? Mit welchen Werkzeugen tauschen sich diese Mitarbeiter aus?

Zur besseren Einordnung und Abgrenzung mit bestehenden SE-Modellen wie das V-Produktmodell oder der XP-Prozess sollten wir nachfragen welche Produkte und Phasen aus diesen Modellen verwendet werden. Generell sollten Antworten darauf gefunden werden ob diese Praktiken durchgeführt werden und wenn wer dafür verantwortlich ist, welche Tools verwendet werden und welche Dokumente anfallen.

- Agile Softwareentwicklung: XP-Praktiken
- The Planning Game — quickly determine scope of next release
- Small releases — put a simple system in production quickly then release new version on a short cycle
- Metaphor — guide development with a simple shared story
- Simple design — system should be as simple as possible, complexity should be removed if at all possible
- Testing — continually write unit tests, customers write functional tests
- Refactoring — restructure the system without changing behavior
- Pair programming — all code written with 2 programmer at 1 machine
- Collective ownership — anyone can change code anywhere anytime
- Continuous integration — integrate and build many times a day
- 40-hour week — work no more than 40h/wk as a rule
- On-site customer — include a real, live user on the team full time
- Coding standards — code in accord. to rules emphasizing communication
- SE-Modell
- Analyse: Werden Anforderungen systematisch / formal aufgeschrieben und verwaltet?
- Design: Gibt es einen (Fach-)Entwurf der Software (z.B. mittels UML)?
- Implementierung:
- Testen: Wird das Produkt systematisch getestet?
- Verifizieren: Wird das Produkt verifiziert (Inspektion, Audit, ...)?

- Evolution: Werden Änderungen am System systematisch dokumentiert?
- Projektmanagement: Werden Projektentscheidungen dokumentiert?
- Qualitätsmanagement: Werden Messdaten über das Produkt, Projekt oder die Organisation erhoben?

4.2.1 Interviewleitfaden: Erfassung Wissenskultur/Wissensaustausch (Schritt 2a)

Beschreiben Sie Erfahrungen oder andere Dokumente welche Sie selbst nutzen würden.

Dauer: ca. 30 Minuten

Material:

Methode: Gruppendiskussion

Ziel: Diese Frage dient dazu, einen reinen Überblick über den State of Practice des Wissensaustauschs in Ihrer Organisation zu erhalten. Ebenfalls die Identifikation und Gewichtung der Wissens- und Erfahrungsquellen um später gezielt und mit größtmöglichem Einfluss Verbesserungen einzubringen.

Anmoderation: Der betrachtete Ausschnitt umfasst dabei das aus 4.1 ermittelte Prozessmodell.

Weitere Fragen zu den Reuse-Abläufen:

- In welchen Phasen fallen „interessante“ Erfahrungen/Artefakte ab? Bei welchen Rollen?
- Wer benötigt welche Erfahrungen/Artefakte in welcher Phase?
- Wer würde die Erfahrungen oder Dokumente produzieren und konsumieren? Die gleichen Personen (-gruppen)?
- Wie weit wäre typischerweise der zeitliche Abstand zwischen Produktion und Konsumierung?
- Wie sieht der Wissenstransfer zwischen Rollen (Tester --> Entwickler, Tester -> Qualitätsmanager) momentan aus?

Zur gezielten Unterstützung der Nutzer von RISE müssen wir Produkte ermitteln mit denen die Benutzer in Berührung kommen und aus denen Informationen zur besseren Benutzung von RISE extrahiert werden können.

- Gibt es typische Vorgehensweisen bei der Wiederverwendung?
- Gibt es bekannte Anlaufstellen (Experten, Kaffeeküche, Foren, Mailing-Listen, etc.) bei denen man sich schnell informieren kann? (Ziel: Erstellung einer Liste von internen und externen Wissensquellen insb. auch für das externe Information Retrieval)

- Gibt es wiederkehrende Probleme bei der Auffindung von notwendigem Wissen?
- Welche Tools gibt es um Projekt-Informationen oder Erfahrungen zu verwalten (z.B. Projekt-Folder, Lotus, WIKIs, Excel / Access Datenbanken)? (Ziel: Cross-Check mit den Wissensarten, Bezug Erfahrungsarten / Tools)

4.2.2 Interviewleitfaden: Charakterisierung vorhandener Artefakte (Schritt 2c)

Beschreiben Sie die Arten von Erfahrungen oder Dokumente welche Sie selbst nutzen (würden).

Dauer: ca. 30 Minuten Material: eMail Methode: Artefaktanalyse

Ziel: Diese Frage dient dazu, einen reinen Überblick über das Wissen und die Artefakte in Ihrer Organisation zu erhalten. Ebenfalls die Identifikation und Gewichtung der Wissensthemen und Artefakte um später gezielt und mit größtmöglichem Einfluss Verbesserungen einzubringen.

Anmoderation: Der betrachtete Ausschnitt umfasst dabei das aus 4.1 ermittelte Prozessmodell.

Weitere Fragen zu den Artefakten:

- Welche Arten von Artefakten gibt es in den Datenspeichern des Unternehmens? (z.B. Produktbeschreibungen, Projektbeschreibungen, Anleitungen, Erfahrungen, Kommentare, Meinungen, Change Request-Listen, Open Issues, Teams/Abteilungen), ...)
- Welche Arten von Erfahrungen oder Wissen wird zwischen den Mitarbeitern ausgetauscht? (z.B. Tipps, Erfahrungen, News, ...)
- Welche Themen / Topics sind Inhalt der Erfahrungen und Artefakte? (Programmiersprachen (Java, C++), Technologien (J2EE, .NET), Datenbanken (mysql, oracle), Tools (CVS, Clover, ...), Standards (XML, RDF, ...) , Kunden, Konkurrenten, ...)

Zur gezielten Unterstützung der Nutzer von RISE müssen wir Produkte ermitteln mit denen die Benutzer in Berührung kommen und aus denen Informationen zur besseren Benutzung von RISE extrahiert werden können.

- Gibt es typische Vorgehensweisen bei der Anpassung von Artefakten oder Erfahrungen?
- Gibt es bekannte Anlaufstellen (Experten, Kaffeeküche, Foren, Mailing-Listen, etc.) bei denen man sich schnell informieren kann?

- Gibt es wiederkehrende Probleme bei der Suche, dem Auffinden, dem Verstehen oder beim anpassen von Wissen oder Artefakten?

4.2.3 Interviewleitfaden: Charakterisierung der WM-Tool-Infrastruktur (Schritt 3)

Beschreiben Sie die WM-Tools die zur Kommunikation zwischen den Mitarbeitern eingesetzt werden.

Dauer: ca. 60 Minuten Material: Methode: Einzelinterview

Ziel: Diese Frage dient dazu, einen reinen Überblick über die Werkzeuge und die Wissensstrukturen in Ihrer Organisation zu erhalten. Ebenfalls die Identifikation von Problemen mit den Werkzeugen und Strukturen um später gezielt und mit größtmöglichem Einfluss Verbesserungen einzubringen.

Weitere Fragen zu den Tools:

- Wer hat das wie WIKI eingeführt? (Gab es eine Einführung oder einen Ansprechpartner? Gab es einen Stakeholder / Vorbild / Knowledge Champion? Eine bestimmte Abteilung oder Projektgruppe?)
- Wurde es von allen akzeptiert?
- Warum wurde das WIKI eingeführt? (Gab es eine Notwendigkeit? Gab es eine klare Zieldefinition? Wissensverlust?)
- War die Administration des WIKI's aufwändig?

Zur gezielten Unterstützung der Nutzer von RISE müssen wir Produkte ermitteln mit denen die Benutzer in Berührung kommen und aus denen Informationen zur besseren Benutzung von RISE extrahiert werden können.

- Gab es genügend Zeit neben der Arbeit Inhalte zu generieren? (Zeitmangel, Projektstress zu groß?)
- Welche Anreize gab es Inhalte zu generieren? (Zeit, eigener Nutzen, Anerkennung, Geld, ...)
- Wann wurde das Wiki benutzt? (Arbeitsbegleitend (d.h. anstelle eines gelben Zettels), Retrospektiv (zur Nachdokumentation von Projekten, Post Mortem), Prospektiv (zur Planung von Projekten und Dokumentation von Zielen / Erwartungen?)

5 Ergebnisse zur Kontextermittlung

5.1 Probleme bei der Wikinutzung (Schritt 0b)

5.1.1 Gruppendiskussion bei der brainbot technologies AG

Im Mittelpunkt der Diskussion stand der Einsatz eines WIKIs für die Unterstützung der Arbeit beim Anwendungspartner brainbot technologies AG.

Kernpunkte der Diskussion bei der brainbot technologies AG

- Wichtige Informationen über das Softwaresystem stehen im Versionierungssystem für die Software (hier: Subversion).
- Konzepte und Dokumentation der Software standen teilweise auch im WIKI
- Beschreibungen der Funktionen einer Komponente befinden sich direkt im Quellcode in einem für ePyDoc lesbaren Format (vergleichbar mit Javadoc).
- Änderungen am Softwaresystem werden per eMail (d.h. mittels der Change-log Funktion von CVS) an alle Interessierten verschickt.
- Datenquellen (d.h., Repositories) mit projektrelevanten Informationen sind: Projektfolder, Netzlaufwerke, eMails, Task Cards (am Boardboard), WIKI
- Inhalte des WIKI's waren: Kundeninfos, Termine, TODO-Listen, Adressen, Konzepte zu Softwarekomponenten, Fehl- und Anwesenheitslisten.

Was läuft gut?

- Die Kooperation zwischen zwei eng zusammenarbeitenden Mitarbeitern lief gut.

Was läuft schlecht?

- Weder ePyDoc noch WIKIs erlauben eine (einfache) Integration von Bildern oder Graphiken.
- Die Suche nach Seiten bzw. Informationen im WIKI ist ungenügend.
- Bei der Suche konnten die CamelCases in den Seitennamen nicht aufgelöst werden.
- Die Umbenennung von Seiten im WIKI ist kompliziert.

- Das Auffinden von Informationen ist kompliziert da diese über mehrerer Repositories verteilt sind.
- Zugriff aus Text-Editoren (z.B. emacs, oder eine shell) ist unkomfortable bzw. unmöglich

Vermutung für Ursachen der Probleme:

- Die Benutzung und Wartung des Wikis ist Freizeit, es gibt dafür keinen expliziter **Aufwand**

5.1.2 Gruppendiskussion bei empolis

Im Mittelpunkt der Diskussion stand der generelle Umgang mit Wissen im Rahmen der Software-Entwicklung von Software bei empolis. Dabei wurde insbesondere auf die aktuellen Barrieren und Herausforderungen und die jeweils eingesetzten Tools eingegangen.

Kernpunkte der Diskussion bei Empolis

- Das Wiki wurde initial stark genutzt, Nutzung geht jetzt zurück wg. **Unübersichtlichkeit** der Inhalte.
- Eine **Topic map** des Wikis wurde von Hand angelegt. Generell könnten solche Übersichten den fehlenden Überblick herstellen, zumindest aber als Tools für den Gardener dienen.
- Das Wiki ist **keine KM Base!** Es wäre aber nicht schlecht wenn die Mitarbeiter mehr von ihrem Wissen rein schreiben würden.
- Es stehen **keine Lessons Learned** drin (z.B. Release x.y, WebServer 3.2 und Modula.b. funktioniert nur mit Compiler Version 1.1). Dies wäre aber wünschenswert, erfordert aber eine gute Suche über Problem- oder Produktversionsklassen.
- Es ist keine Einführung erfolgt, alle wurden gebeten, ihre Homepage (Beschreibung Person & Skills) im Wiki als Probestück anzulegen. Dies wurde von der Geschäftsleitung gefordert.
- Wiki wird oft als Vorwand genutzt, um scheinbar auf Informationsanfragen einzugehen: **„Es steht im Wiki“** (Einwand der Dokumentations-Verantwortlichen)
- Es existieren mindestens zwei Wikis: MASE – als allgemeiner Wiki, und im Rahmen eines „geheimen“ Projektes, noch ein JSP Wiki, um hier die **Dokumentation** zu erstellen. Da in diesem Projekt explizit Ressourcen für die Redaktion von Inhalten zur Verfügung gestellt wurden, wuchert dieser Wiki nicht so stark und wird als nützlich angesehen.

- Wikis werden u.A. genutzt, um **Anforderungen** zu erörtern. Dabei werden Wikis als Diskussionsforum „missbraucht“. Eine Trennung von eigentlicher Wiki Seite und Diskussion wäre laut Aussage eines Teilnehmers – schön.
- Es existiert eine Log-Liste der Änderungen.
- (Mit) ein Grund für die Einführung des Wikis war, dass die Administration des **Exchange-Servers** auf einer Volumenbasis outsourced wurde. Die öffentlichen Verzeichnisse in Exchange wurden daher abgeschaltet, um Kosten zu senken.
- Verwendete **Tools**: CVS, outlook/exchange, office, messenger, fileserver, omega, Techdirector, NetMeeting, icq, web-ex
- Verwendete **Wissensquellen**: MSDN, Google

Was läuft gut?

- Das Wiki ist über VPN von außen verfügbar und alle können es benutzen (auch von zuhause)
- Vor jeder Neuentwicklung eines Produktes, wird solange diskutiert bis alles klar ist. Erste Spezifikationen (hauptsächlich auf technischem Level) werden auch über das Wiki erstellt.

Was läuft schlecht?

- Die **Seitenanzahl** im WIKI explodiert
- **Strukturierung** ungenügend, jeder strukturiert für sich selbst
- Strukturen werden von allen erstellt und geändert. Dies erzeugt viel Verwirrung bei den Mitarbeitern.
- Man kann sich die **Links** sehr schwer merken
- **Suche** unzureichend, man findet nur eigene Sachen (die man kennt) oder auf die Dinge, auf die man Hingewiesen wird.
- Benutzung als Write only memory; man nichts findet
- Fehlende oder zu umfangreiche **Notifikationen**
- Wiki muss die Bearbeitung ad hoc beschleunigen, ein **WYSIWYG** wäre hilfreich
- Es wurde erwähnt, dass man damit keine **Worddokumente** bearbeiten kann
- Fehlende Klarheit über den Ort der **Informationsablage**: Wann das Wiki nehmen, wann ein anderes Werkzeug?

- Es gibt keine direkte **Zuordnung** zwischen Releases und Wikiseiten. Bei der Erstellung von Dokumenten, die an den Kunden gehen, wird diese Information auf der Wikiseite hinzugefügt.
- Die **Editierung** mit dem Wiki dauert länger als mit E-Mail, bei Zeitdruck wird auf **Email** zurückgegriffen.

Vermutung für Ursachen der Probleme:

- Die Benutzung und Wartung des Wikis ist Freizeit, es gibt dafür keinen expliziter **Aufwand**
- Fehlende **Co-Integration** mit anderen Systeme (SW-Entwicklung, Administration), z.b. Filesystem, etc.
- Wiki ist nur ein "**weiteres System**" in dem man suchen muss.
- Fehlende **Kommunikationsregeln**: Es ist nicht klar, wo etwas steht – so muss man in unterschiedlichen Systemen suchen, es dann doch nicht zu finden.
- Fehlende **Informationslogistik** / Interessenskonflikt bei der Informationsbereitstellung: Oft scheint nicht klar zu sein, wer welche Information in welcher Form benötigt. Es hat den Anschein, dass teilweise auf Verdacht oder aus Tradition dokumentiert wird. Wenn dies klar ist, dann muss Person A für eine Person B Informationen erfassen, die Person A nicht für sich erfassen müsste.

Eigener Eindruck:

- Die SE / XP spezifischen Features von MASE werden nicht genutzt (z.B. Task planning) Grund ist vermutlich, dass es bereits andere Systeme gibt. Wenn dies der Fall ist, dann benutzen sie eigentlich JSP-Wiki. Evtl. wäre noch zu klären, ob der Bewertungs-Plugin oder das CVS plugin genutzt wird.

5.2 Erfassung des Software Entwicklungsablaufs (Schritt 1a)

5.2.1 Gruppeninterviews bei empolis

Im Rahmen eines Interviews mit dem Entwicklungsleiter und einem Produktlinien-Manager wurden die im folgenden beschriebenen Teilprozesse erfasst.

Zuerst werden die Rollen beschrieben, und daran anschließend die Prozesse vorgestellt:

5.2.1.1 Rollenmodell

Im folgenden werden die für die Entwicklung relevanten Rollen – basierend auf dem oben genannten Diagramm – bzgl. Ihre Aufgaben und Kompetenzprofilen weiter detailliert. Weiterhin werden, wenn möglich, Vertreter dieser Rollen genannt.

Entwicklungsleiter

Der Entwicklungsleiter ist verantwortlich für die Software Entwicklung bei empolis. Zu seinen Aufgaben gehört daher unter anderem die allgemeine Projektkoordination und Verbesserung der Software-Entwicklungs-Infrastruktur.

Der Entwicklungsleiter hat zudem einen Überblick über die Skills der Entwickler. Über das Tool Help4Project kann er sich einen Überblick über die aktuelle Auslastung der Entwickler verschaffen.

Entwickler

Neben der Codierung selbst sind die Entwickler für die Code-Dokumentation und das Testen auf Modul-Ebene verantwortlich. Sie können im Rahmen einer Akquise nach Abschätzungen über zu erwartende Aufwände gefragt werden. Weiterhin diskutieren die Entwickler die Anforderungen an die Software im Wiki.

Die meisten Entwickler haben einen Universitätsabschluss in Informatik. Die Know-How Profile sind technologisch differenziert (z.B. Applikationsserver, Datenbanken). Bei den Programmiersprachen herrscht Java vor, es werden aber auch C++/C# eingesetzt.

Projekt-Manager

Der Projekt-Manager ist der Budget-Verantwortliche für ein Projekt und die Abwicklung des Projekts. Er weist die Arbeitsaufträge den im Projekt beteiligten Entwicklern zu. Der Projekt-Manager ist in der Regel selbst ein Entwickler im Projekt. Bei der Akquise wird in der Regel der zukünftige Projektmanager als technischer Ansprechpartner eingeschaltet.

Der Projekt Manager verfügt über die vergleichbare technischen Skills wie ein Entwickler. In der Regel ist der Projekt-Manager schon länger bei empolis angestellt und hat zumindest einen Überblick über die Organisation und Arbeitsabläufe.

Dokumentations-Spezialist

Die Dokumentations-Spezialisten erstellen die über der Code Ebene liegende technische Dokumentation der Software.

Für Ihre Arbeit nutzen sie das Wiki, das Projekt-Verzeichnis oder kommunizieren direkt mit den jeweiligen Informationsträgern.

Produktlinien-Leiter / Solution Manager

Der Produktlinien-Leiter ist für koordiniert Projekte in seinem Bereich, um eine Fortentwicklung über Kundenprojekte zu ermöglichen. Dazu gehört insbesondere, die Erstellung von der halbjährlichen Releases zu koordinieren.

Der Produktlinien-Leiter ist daher in der Regel der Projektmanager für die Projekte, die sich in seiner Produktlinie abspielen.

Sales

Die Sales-Vertreter sind die Hauptverantwortlichen für die Akquise neuer Projekte und die Pflege von Bestandskunden.

Sie haben Kenntnis über die Anwendungsdomäne, allerdings ist die Kenntnis über technischen Details gering. Aus diesem Grund besteht in der Regel ein Akquiseteam aus mindestens einer Sales-Person und einem technischen Vertreter.

Help-Desk Support

Der Help Desk Support ist die erste Anlaufstelle für Kunden und Kollegen bei Problemen mit der Software. Insbesondere werden hier noch nicht gefundene Bugs als solche erkannt und dann der Entwicklung weitergegeben.

5.2.1.2 Prozesse

Die im folgenden beschriebenen Prozesse beziehen sich – sofern nicht anders angegeben - auf die Vorgehensweise am Standort Kaiserslautern.

Akquise

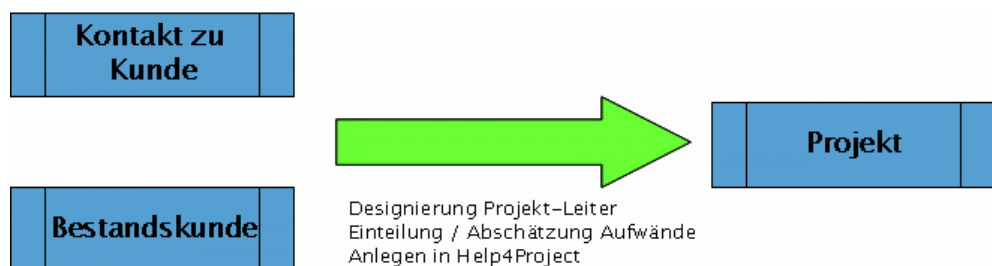


Bild 3

Ablauf Akquise

Die Erstellung eines Angebots wird initial von Sales getrieben. Dabei wird möglichst früh die technische Seite in Form des designierten Projekt-Leiters (oft der jeweilige Produktlinien-Manager) mit einbezogen, um das vier Augen Prinzip zu gewährleisten. Sobald projekt-spezifische Aufwände verbucht werden müssen, wird das Projekt in Help4Project angelegt.

Auftragserteilung, Abrechnung und Projektdurchführung aus Control- lingsicht

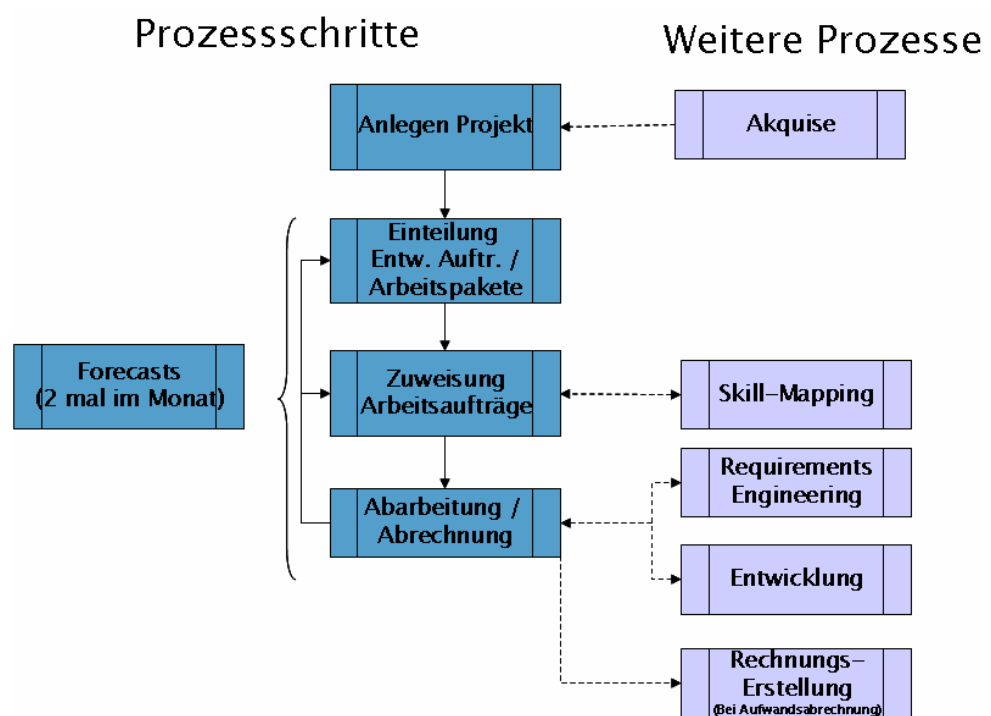


Bild 4

Ablauf Auftragserteilung, Abrechnung und Projektdurchführung mit Bezug zu anderen Prozessen

Dieser Prozess wird durch das Tool Help4Project unterstützt. Daher wird für diesen Prozess in erster Linie die Verwendung des Tools beschrieben.

Den ersten Schritt stellt das Anlegen eines Projekts in Help4Project – in der Regel im Rahmen einer Akquise (Neukunde oder Bestandskunde) dar. Darauf aufbauend werden für die Projekte Arbeitspakete definiert und daraus die Entwicklungsaufträge abgeleitet. Diese Arbeitspakete werden den einzelnen Mitarbeitern von empolis basierend auf Ihren Fähigkeiten zugewiesen. Bei der Abarbeitung / Abrechnung erfasst das Tool - als auswertbare Daten - die Aufwandsplanung und Tätigkeitsabrechnung auf diesen Arbeitspaketen. Diese Daten geben die geleisteten Arbeiten im Rahmen der Entwicklung und des Requirements-Engineerings wieder. Zusätzlich werden die Start und Enddaten eines Projekts

mit erfasst. Es gibt aber keine automatisierbare Möglichkeit die, Start und Endpunkte eines Arbeitspakets zu bestimmen.

Sobald auf ein Projekt Aufwände verbucht werden, taucht es in diesem System auf (z.B. im Rahmen der Akquise). Daher kann man am Zeitpunkt der Einstellung den Beginn der projektspezifischen Akquise-Phase eines Projektes erkennen. Als Endpunkte der Projekte kann der Abschluss des Buchung von Aufwänden im letzten Arbeitspaket eines Projektes genommen werden.

Das Tool, das diesen Prozess unterstützende, wird umfassend akzeptiert und eingesetzt. Nur wenige Projekte werden nicht über dieses Tool erfasst. Dieses Tool erfasst auch Aufwände für Overhead. Das vergleichbare Task-Planungssystem in MASE wird nicht genutzt. Die hier verfügbaren Einträge sind auf die Erprobungsphase zurückzuführen.

Das Tool verfügt über zwei grundlegende Sichten: Für Budgetverantwortliche und Mitarbeiter. Die Budgetverantwortlichen (Projekt-Manager, Teil-Projekt-Manager) haben Zugriff auf die Zuweisung von Aufwänden an Mitarbeiter und eine Übersicht über Soll / Ist Stand. Neben den Budgetverantwortlichen hat noch Führungspersonal Zugriff auf diese Daten (z.B. Entwicklungsleiter Mario Lenz). Die Mitarbeiter können hier ihre auf den jeweiligen Arbeitsaufträgen geleisteten Stunden verbuchen. Diese Zuweisung erfolgt auf Stundenebene.

Skill Mapping zu Arbeitsaufträgen

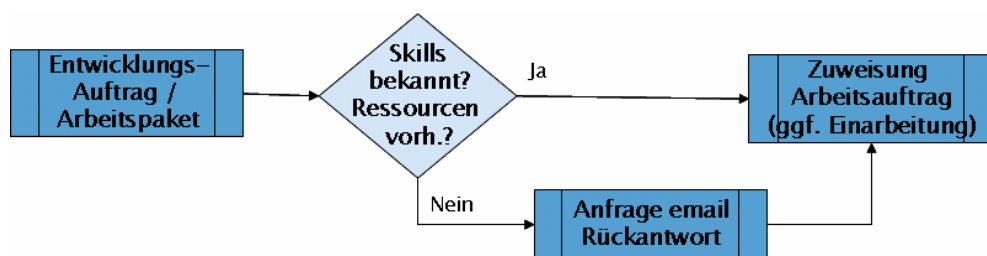


Bild 5

Ablauf Skill-Mapping

Die Zuweisung zu Arbeitsaufträgen basiert auf der Einschätzung der Eignung der jeweiligen Person durch den Entwicklungsleiters oder Teamleiters. Da es sich um ca. 24 Personen handelt, ist diese erfahrungsbasierte Zuweisung machbar. Als Kriterien für die Eignung kommen die Erfahrung mit den jeweils eingesetzten Technologien (z.B. Programmiersprachen, eingesetzte Komponenten wie Datenbanken oder Applikationsserver), das Problemlösungs-Verhalten und die generelle Fähigkeit zur Einarbeitung zum Einsatz. Falls eine Kompetenz nicht vorhanden ist, sind die Entwickler durch Ihre Ausbildung in der Lage, sich einzuarbeiten. In seltenen Fällen werden bestimmte Aufgaben an Dritte vergeben.

Falls nicht abzusehen ist, ob eine Kompetenz vorhanden ist, wird eine E-mail an alle Entwickler geschickt, auf die ggf. der jeweilige Kompetenzträger antwortet (Beispiel: Stemming-Komponente für Chinesisch).

Release Management (Requirements-Engineering und Entwicklung)

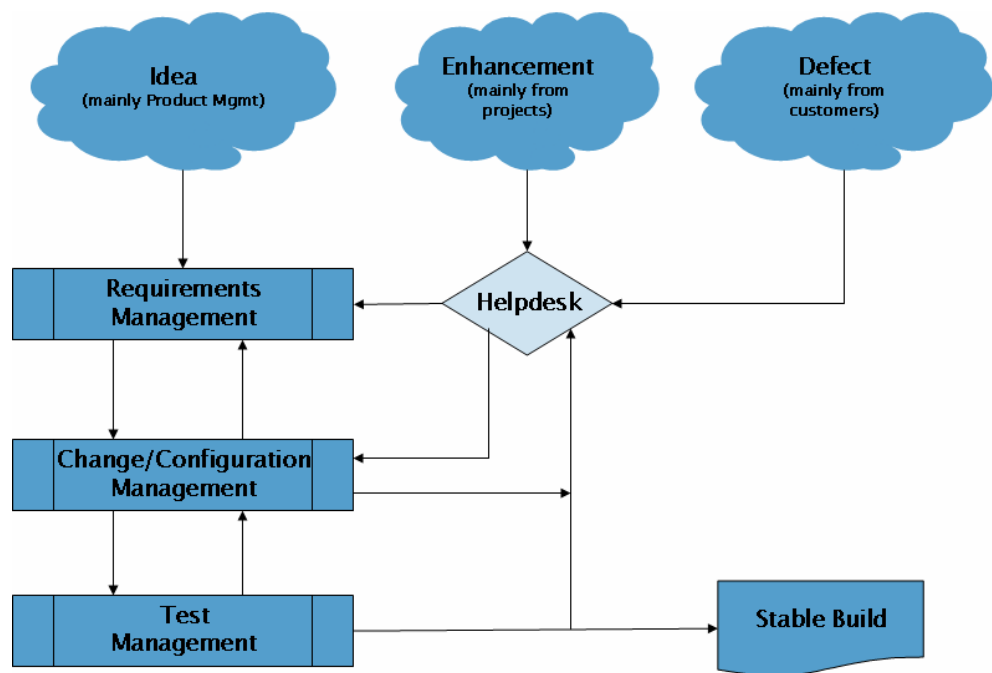


Bild 6

Ablauf Release Management (Kopiert auf Präsentation Mario Lenz vom 14.12.04)

Die obenstehende Grafik gibt eine Übersicht über das Release Management und damit die Entwicklung des Software-Frameworks orange in Kaiserslautern. Da dieses Produkt seit mehreren Jahren auf den Markt ist, werden in der Regel nicht komplett neue Releases entwickelt, sondern vorhandene Funktionalität verbessert oder neue Funktionalität in das Framework hinzuimplementiert. Die Strukturierung stellt den gewünschten Ablauf dar, der im Rahmen der Vereinheitlichung der Prozesse für alle Standorte geplant ist. Auf der oben angegebenen Abstraktionsebene wird dieser Prozess in Kaiserslautern im wesentlichen befolgt. An andern Standorten (Würzburg / Gütersloh) kann es eher zu Abweichungen bei der Definition eines Releases und bei der konkreten Ausgestaltung des Prozesses kommen. Grund ist die kundenspezifischere Lösungsentwicklung und die eingesetzten Tools, die zum Teil einen andern Entwicklungsablauf unterstützen oder Begrifflichkeiten vorgeben.

Ausgangspunkt für das Release Management sind Informationen, die Veränderungen in orange bewirken. Hier kann man im wesentlichen drei Informationstypen unterscheiden: Ideen, Verbesserungen und Defekte.

- **Ideen** sind Verbesserung oder erstmaligen Implementierung einer Funktionalität. Diese Ideen werden in erster Linie in Projekten entwickelt. Eine weitere Quelle für diese Ideen ist die Marktbeobachtung und die so gewonnene Übersicht über die Funktionalitätsunterschiede zu den Mitbewerbern. Diese Ideen werden in der ersten Requirements-Engineering Phase – oft auch unter zu Hilfenahme des Wikis - diskutiert und weiter konkretisiert.
- **Verbesserungen** sind Vorschläge für gewünschte Systemfunktionalitäten von Seiten des Kunden. Diese Vorschläge werden an das Support Team herangetragen, welches über die Aufnahme zur weiteren Bearbeitung entscheidet. In einigen Fällen können solche konkreten Verbesserungen auch durch den jeweiligen Produktlinien-Leiter aufgeworfen werden.
- **Defekte** oder **Bugs** schließlich sind Informationen über Fehlverhalten der eingesetzten Applikation. Ob ein gemeldeter Defekt tatsächlich auf ein Fehlverhalten der jeweiligen Komponente zurückzuführen ist, wird durch den Support entschieden. Je nach Vertragslage stößt diese Meldung kundenspezifische Wartungsaktivitäten an. Wenn es sich dabei um Änderungen am Framework handelt, sollten diese zum nächsten Release in den Hauptentwicklungszweig zurückgespielt werden.

Das Release Management selbst ist – wie in der obigen Abbildung – in drei Phasen unterteilt:

Im Requirements-Engineering werden Anforderungen an das zu entwickelnde System weiter konkretisiert und ggf. in eine Spezifikation überführt. Aus diesen Angaben wird dann ein Change Request erzeugt, welcher in der nächsten Phase abgearbeitet wird. Die Notation der Anforderungen ist abhängig von den Projektbeteiligten.

Im Change / Configuration Management wird über die Umsetzung des Change Requests entschieden und bei Akzeptanz implementiert. Innerhalb dieses Prozesses findet die Implementierung der Software statt. Dabei muss zum einen zwischen der Entwicklung der Kernkomponenten und der kundenspezifischen Teile entwickelt werden. Dabei kann, wenn nötig Pair Programming „auf Zuruf“ eingesetzt werden.

In der Test-Phase werden die zuvor entwickelten Funktionalitäten getestet. Dabei ist zwischen Unit Tests, Integrations- und System-Tests zu unterscheiden. Unit Tests werden vom Entwickler selbst durchgeführt. Integrations-Tests beziehen sich auf einen Build, d.h. den aktuellen (kundenspezifischen) Stand der entwickelten Software. Release Test schließlich validieren ein Release. Der Test-Aufwand erfolgt in Abhängigkeit von den jeweiligen Projektanforderungen. Wenn nötig, wird dazu eine dedizierte Rolle bestimmt. Auch hier kommt in Gütersloh der Test Director zum Einsatz. Neue Releases werden in KL unter anderem durch Anwendung auf bestehende Demo-Systeme validiert.

Dokumentation (Projekt / Produkt)

Die Dokumentation für Produkte wird in KL durch zwei Dokumentationsspezialisten erstellt. Diese nutzen als (einen) Input für Ihre Arbeit das Wiki. Die Erstellung der Produkt-Dokumentation erfolgt im Rahmen des Release-Erstellung.

Die Dokumentation von Projekten ist abhängig vom Standort, als auch von den Anforderungen des Geldgebers. Die standortabhängige Unterscheidung ist wie folgt: In KL wird zur Projektdokumentation der allgemein verfügbare oder ein projektspezifischer Wiki benutzt. In Würzburg wird das IT-Projekt-Management-Framework Prince 2 eingesetzt. Durch dieses Framework werden Templates (z.B. Projektbeschreibung, Change Request) vorgegeben. Zusätzlich können die Anforderungen des Geldgebers Einfluss auf die Projektdokumentation nehmen. Ein Beispiel sind Projekte, die nach dem einem ISO-Standard abgewickelt werden.

5.2.2 Interviews bei der brainbot technologies AG

Zuerst werden die Rollen beschrieben, und daran anschließend die Prozesse vorgestellt:

Salesman und Marketing Manager (SM): Der Salesman und Marketing Manager ist für das Marketing bzw. die Akquise neuer Kundenprojekte zuständig. In der Regel wird der SM bei Kundenkontakten vom GI oder PL unterstützt, da der SM keinen direkten Kontakt zur Software-Entwicklung hat. Der SM besitzt jedoch ein grundlegendes Verständnis von technischen Aspekten. Zu seinen Kernaufgaben zählen außerdem Finanzen und Controlling.

Geschäftsführer und Innovationsmanager (GI): Während auch der Projektleiter gelegentlich Prototypen erstellt, gehört diese Aufgaben zu den Kernaufgaben des Geschäftsführers und Innovationsmanagers. Der GI erstellt häufig Software-Artefakte, auf die sich die spätere Software-Entwicklung stützt. In Abhängigkeit von der Größe des Projekts werden Prototypen direkt weiter verwendet bzw. vorwiegend, um daraus Demonstrationen zu erstellen. Entsprechende Prototypen werden zu einer Reife gebracht, die es potentiellen Kunden erlaubt, wie hoch ein mögliches Projektbudget sein könnte und welche Funktionen des Prototypen überarbeitet bzw. neu erstellt werden müssen. Der GI kommuniziert außerdem mit Geschäfts- bzw. Vertriebspartnern.

Projektleiter (PL): Der Projektleiter ist für die Koordination der agilen Softwareentwicklung zuständig. Der PL erstellt gelegentlich auch Prototypen oder nutzt bestehende Software, um Tests durchzuführen und Demos zu erstellen. Die vom PL erstellten Prototypen dienen jedoch primär einer frühen Machbarkeitsanalyse. Der Projektleiter stellt somit das Bindeglied zwischen dem Kunden - welche bereits in frühen Entwicklungsstadien mögliche Lösungsansätze bewer-

ten möchten - und den Softwareentwicklern dar. Zum Aufgabenbereich des PL gehören auch administrative Tätigkeiten und Wartungstätigkeiten. Hauptentwickler (HE): Der Hauptentwickler ist an der Konzeption der Software-Architektur beteiligt, um von Anfang an konkrete Lösungsvorschläge einbringen zu können. Außerdem liefert er der Projektgruppe wichtige Informationen zur technischen Machbarkeit. Der Hauptentwickler hat dadurch großen Einfluss auf die Anwendungsentwicklung, da nur solche Software-Architekturen umgesetzt werden, welche seine Zustimmung finden. Die Rolle des HE ist vorwiegend beratender Natur, d.h. der Hauptentwickler schreibt selbst keine Dokumente zur Software-Architektur, sondern bewertet diese hauptsächlich.

Entwickler (E): Während der Hauptentwickler die Kernkomponenten der Software entwickelt, erstellt der Entwickler in Absprache mit dem HE periphere Softwarekomponenten. Der Entwickler bringt sich gelegentlich beim Entwurf von Software-Architekturen ein bzw. nimmt an Besprechungen teil.

5.2.2.1 Prozesse

Die im folgenden beschriebenen Prozesse beziehen sich auf die allgemeine Vorgehensweise die bei den meisten Projekten welche die brainbot technologies AG durchführt anzutreffen sind.

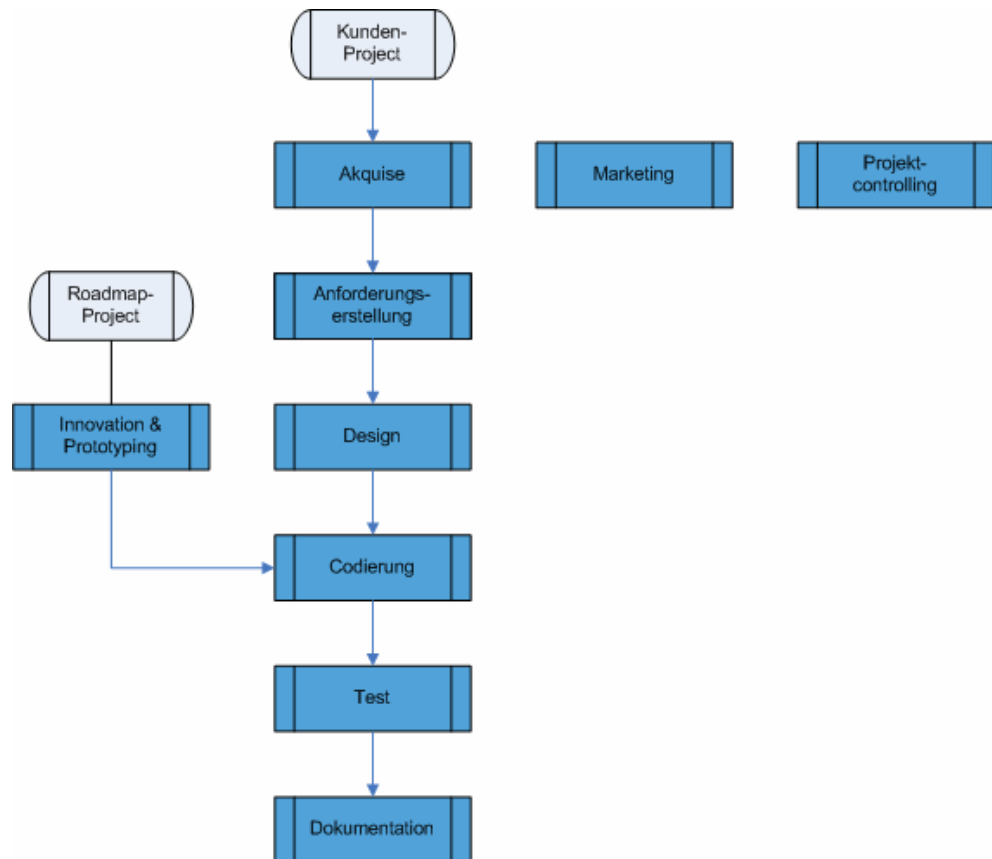


Bild 7 Prozessmodell bei der brainbot technologies AG

Bei der brainbot technologies AG wird grundsätzlich zwischen Roadmap-Projekten und Kundenprojekten unterschieden. Bei Roadmap-Projekten geht es darum neue Technologien zuerst intern, d.h. ohne Kundenanfrage, zu erproben und gegebenenfalls als neue Kernkomponente in das Entwicklungsframework (Bestehenden aus einem Satz Kernkomponenten) aufzunehmen. Bei Kundenprojekten wird unterschieden zwischen Projekten bei denen eine neues Produkt für den Kunden entwickelt werden, und den Projekten bei denen bereits ausgelieferte Produkte gewartet und angepasst werden.

Abbildung zeigt den Ablauf von Kundenprojekten als auch von Roadmap-Projekten. Es können dabei die folgenden Phasen durchlaufen werden: Akquise, Erstellung der Anforderung, Architekturdesign, Programmierung, Test und Dokumentation. Je nach Projekt werden manche Phasen auch übersprungen. Dies kann z.B. der Fall sein bei kleineren Anpassungsprojekten. Auf diese einzelnen Fälle wird später noch eingegangen. Es gibt keine strenge Vorschriften was die Phasenabläufe angeht. Bei der brainbot technologies AG wird eher agil und iterativ entwickelt, d.h. das die Phasen mehrmals während eines Projektes

durchlaufen werden können. Eine klare Trennung zwischen den Phasen besteht grundsätzlich nicht.

Die Phasen werden im Folgenden beschrieben. Vor allem die eigentlichen Entwicklungsphase sind von besonderer Relevanz im Rahmen von RISE. Deswegen werden Marketing und Projekt-Controlling nur kurz beschreiben.

Akquise

Die Akquise führt in den meisten Fällen der Geschäftsführer direkt oder die Salesperson, die auch verantwortlich ist für das Marketing ist, aus. Dabei greifen diese Personen auf gespeicherte Kundendaten im Wiki oder auf den *Company-Ordner* zu. Für jedes akquirierte Projekt gibt es einen Company-Ordner im Dateisystem. Alle Dokumente die während des Projektes entstehen werden dort abgelegt.

Anforderungserstellung

Die Anforderungserstellung ist für die Projekte unterschiedlich. Die Personen die auch für die Akquise zuständig sind entwickeln auch die Anforderungen. Dabei gibt es unterschiedliche Formalitätsgrade. Die Anforderungen werden zuerst teils semi-formal über Use-Cases beschrieben, oder informal mittels Wikiseiten. Wenn die Anforderungen mit dem Kunden abgestimmt worden sind, wird in Microsoft Word ein Pflichtenheft erstellt, der wiederum im Company-Folder abgelegt wird. Grundsätzlich ist der Kunden bis in dieser Phase eng eingebunden. Bei den nachfolgenden Phasen ist der Kundenkontakt geringer. Zur Kommunikation mit dem Kunden werden vor allem das Wikis als auch Emailverkehr verwendet.

Design

Allen Entwicklungsprojekten liegt ein Komponenteframework zugrunde. Das Framework besteht aus mehreren Kernkomponenten die immer wieder in den verschiedenen Kundenprojekten verwendet werden. Der Architekturdesign des zu entwickelnden System besteht also darin, benötigten Kernkomponenten die schon vorhanden sind auszuwählen und wieder zu verwenden. Dokumente die die Architektur beschreiben werden meistens nicht angefertigt. Vor der Festlegung des Architekturdesign, erstellen die beteiligten Entwickler informell eine Architektur am Whiteboard. Während dieser Phase brainstormen die Entwickler über die zu realisierende Architektur. Wenn ein Design dokumentiert wird, dann wird keine formale anerkannte Beschreibungssprache verwendet. Photoshop wird zur Visualisierung der Architektur verwendet, während diese im Design-Dokument näher erläutert wird.

Innovation und Prototyping

Wie oben schon beschrieben, werden neben Kundenprojekten auch Roadmap-Projekte intern durchgeführt. Ein Entwickler testet dabei neue Technologien und überprüft deren Nützlichkeit mit Bezug auf zukünftige Kundenprojekte, oder Vorteile für das Framework. Entscheidungen werden in dieser Phase getroffen, ob es sich lohnt neue Kernkomponenten zu entwickeln. Um die Nützlichkeit zu prüfen, werden Prototypen entwickelt. Dabei wird keine weitere Dokumentation als der Code selbst erstellt. Es geht allein darum das Potential dieser neuen Komponenten mit minimalem Aufwand zu analysieren.

Fällt eine positive Entscheidung bzgl. der neuen Komponente so entwickelt der Hauptentwickler die Kernkomponente neu und bettet sie in das Framework ein. In den meisten Fällen wird der Code der bei dem Prototyping produziert wurde nicht wieder verwendet.

Kodierung

Die Kodierung erfolgt in den meisten Fällen von dem Hauptentwickler und einem weiteren Entwickler. Die Entwicklung erfolgt mit C++, Python und manchmal mit Delphi für die GUI. Java wird nur für Prototypen verwendet. Die Entwicklung erfolgt grundsätzlich mit dem Emacs.

Die Kodedateien werden mit *Subversion* verwaltet und der Issue Tracker *Roundup* verwaltet die zu erledigenden Aufgaben der Entwickler. Die Verfolgbarkeit innerhalb des Entwicklerteams während der Codierung erfolgt über Change-logs (mit Subversion) und dem Issue Tracker Roundup.

Wenn der Wunsch besteht, dann bekommt der Kunde auch einzelne Change-logs zugesendet oder er bekommt Zugriff auf die Issues über die ZOPE-Plattform. Am Ende eines Tages werden alle Änderungen die am System vorgenommen wurden per Email an alle Entwickler versendet.

Test

Das Testen erfolgt eher ad-hoc und nicht systematisch. Auf Komponentenebene werden die Testfälle oft anhand von Bugsreport erstellt, d.h. das vor allem Codefragmente die in der jüngsten Vergangenheit Probleme bereitet haben getestet werden. die Testfallüberdeckung ist also eher gering.

Auch der Systemtest vor der Auslieferung des Produktes verläuft ad-hoc. Es werden vor dem Test informell Nutzerszenarien ausgedacht und direkt am System durchgeführt.

Dokumentation

Die Dokumentation hat im allgemeinen bei der Softwareentwicklung einen nebenläufigen Relevanz. Wenn es nötig erscheint, dann dokumentiert der Hauptentwickler wichtige Änderungen direkt im Code.

Explizite Dokumentation zum Produkt wie z.B. Dokumentation für die Administration des Systems oder Benutzer Dokumentation werden nur dann erstellt, wenn es explizit vom Kunden gefordert wird. Im Regelfall wird die Dokumentation in einem Word-Dokument fest gehalten. Bei einer nicht-browserbasierten Applikation wurde auch eine Windows-Hilfe Datei ausgehend von HTML-Dateien generiert.

Grundsätzlich wird versucht die Dokumentation konsistent zum entwickelten System während der Entwicklung zu halten. Dazu gibt es aber keine formalen Vorgehensweise.

Marketing

Das Projektkontrolling führt der Manager/Salesman durch. Dabei werden Flyer erstellt und Ideen für das Marketing im Wiki ausgearbeitet.

Projektcontrolling

Das Projektkontrolling führt der Manager/Salesman durch.

5.3 Erfassung der Wissenskultur (Schritt 2a und 2c)

5.3.1 Untersuchung bei der brainbot technologies AG

5.3.1.1 Persönliche Homepages

Die Möglichkeit, eigene Homepage im Intranet zu pflegen, wird nur vom Chefentwickler genutzt. Die Webseite dient allen Mitarbeiter als rudimentäres und zentrales Linkverzeichnis mit den Rubriken „Dokumentationen“, „Mailing-Lists“, „Links“ und „Work“ (Issue Tracker, WIKI etc.).

5.3.1.2 Ablagestrukturen in Email

Bei der Untersuchung der Ablagen in den Email-Clients zeigte sich deutlich, dass diese rollenspezifisch gestaltet werden.

Die Email-Ablagestrukturen wurde häufig kaum bis gar nicht hierarchisch strukturiert. Dies wurde dadurch begründet, dass geeignete Suchtechnologien zum

Wiederauffinden bestimmter Emails zur Verfügung stehen. Deswegen ist aus Sicht der Mitarbeiter der zeitliche Aufwand zur Pflege dieser Strukturen nicht gerechtfertigt.

Die Email-Ablagen wurden teilweise an bereits bestehende Strukturen (Company-Ordner) angelehnt. Diese Strukturen werden jedoch zum Teil nicht mehr genutzt und liegen inzwischen brach. Newsletter bzw. andere nicht-personenbezogenen Mails werden häufig bewusst aufbewahrt, weil die Hemmschwelle zu hoch erscheint, diese zu löschen bzw. weil man den Inhalt als potentiell relevant einschätzt.

5.3.1.3 Company-Ordner

Der sog. Company-Ordner dient bei der brainbot technologies AG als zentrale Ablage auf Filesystem-Ebene. Der Company-Ordner enthält u.a. die folgenden Verzeichnisse: „Akquise“, „Administration“, „Content“, „Finanzen“, „Konkurrenz“, „Management“, „old“, „Papers“, „Projects“, „Testdaten“, „tmp“, „Verträgen“, „Vorlagen“.

Zum Zeitpunkt der Befragung stellte die Aktivität im Company-Ordner die Aktivitäten im WIKI in den Schatten. Alle für das Unternehmen relevanten Informationen wurden früher oder später im Company-Ordner abgelegt. Die Mitarbeiter bei der brainbot technologies AG konnten also davon ausgehen, dass sie die wichtigsten Informationen dort finden können.

Eine wesentlicher Nachteil des Company-Ordners ist jedoch, dass er den von Entwicklern relativ wenig genutzt wurde. Offenbar finde diese im Company-Ordner kein für sie nützliches Material.

Wissenskultur/Wissensaustausch

Die folgende Abbildung zeigt die einzelnen Phasen und die Artefakte die bei der Durchführung der Phasen entstehen. Zu den einzelnen Phasen wurde nun die einzelnen Rollen mit den jeweiligen Kürzeln zugewiesen:

- Salesman und Marketing Manager, Kürzel **SM**
- Geschäftsführer und Innovationsmanager. Kürzel **GI**
- Projektleiter, Kürzel **PL**
- Hauptentwickler, Kürzel **HE**
- Entwickler, Kürzel **E**

Eine Zuweisung in grüner Farbe beschreibt die hauptverantwortliche Rolle für diese Phase. Weiter involvierte Rollen werden durch gelbe Kreise markiert.

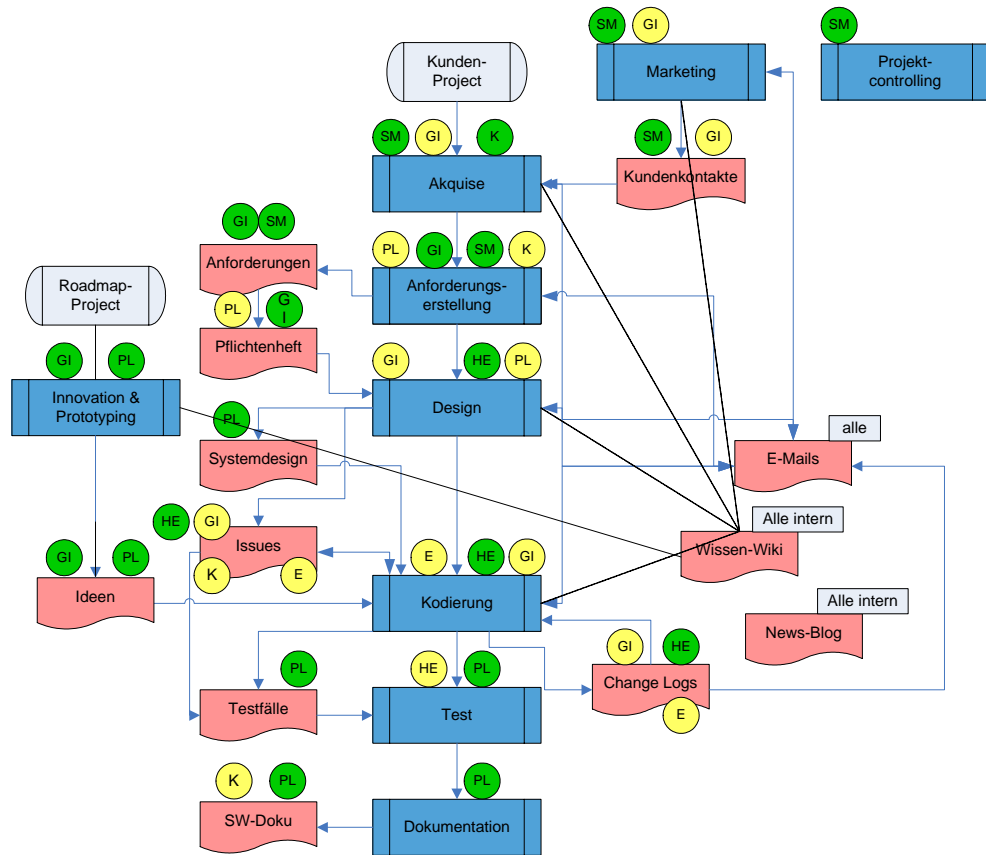


Bild 8 Entwicklungsprozess bei der brainbot technologies AG

Die Abbildung zeigt deutlich, dass in fast allen Phasen nahezu alle Rollen beteiligt sind. Dies ist vor allem der Fall für die Erstellung der Anforderungen, Design und Kodierung. Die Phasen Marketing werden. Andere Phasen wie z.B. Innovation & Prototyping, Marketing, und Projekt-Controlling werden nur von wenigen Rollen belegt. Das Gleiche gilt für die aus den Phasen resultierenden Artefakte.

Außerdem ist aus der Abbildung erkennbar, dass alle Rollen unter einander in fast allen Phasen (außer Testen, Dokumentation und Marketing) vor allem mittels E-Mails kommunizieren. So erfolgt nicht nur die externe Kommunikation mit den Kunden (Kundenakquise und Marketing) per E-Mail sondern auch ein Großteil der Information aus der Software Entwicklung werden per E-Mail verteilt. So werden z.B. jeden Abend die Changelogs per E-Mail an alle Entwicklungsbeteiligten automatisch verschickt. Auch der Issue Tracker benutzt E-Mails zur direkten Informationsvermittlung (per Benachrichtigungs-Service). Zwei weitere zentrale Knotenpunkte für den internen Wissensaustausch sind das Wiki und ein Blog für allgemeinen Nachrichten. Im Blog werden meistens unwichtige Information gestellt, die für die Entwicklung weniger Relevanz haben. Das Wis-

sen im Wiki ist eine zentrale Anlaufstelle sowohl für die eigentliche Software Entwicklung als auch für die anderen Phasen. Hier wird Wissen auf eine informelle Art, ohne fest vorgegebene Strukturen oder Vorlagen abgelegt. Leider werden hier hauptsächlich nur Information zu Kunden, Produkten und allgemeine Information über die reine Büroarbeit hinweg eingestellt - relevante Erfahrungen die während der Entwicklung gemacht werden, werden nur in den seltensten Fällen abgelegt. Wissen über die Softwareentwicklung wird entweder undokumentiert, also verbal in der Entwicklungsgruppe verteilt oder man findet Informationen im Code. Zum größten bleibt dieses Wissen aber implizit, also undokumentiert. Die Information im Code werden hauptsächlich vom Hauptentwickler eingetragen. Die eigene spätere Wiederverwendung spielt dabei einerseits eine große Rolle. Andererseits hat der Hauptentwickler Interesse daran, wenn weitere Entwickler in die Lage versetzt sind, den von ihm bearbeiteten Code verstehen und erweitern zu können.

Im Vergleich zum Quellcode-Dokumentation werden Erfahrungen hingegen nur sporadisch dokumentiert. Als mögliche Ursachen hierfür wurden genannt: die Unbewusstheit, dass eine für spätere Projekte wichtige Erfahrung gemacht wurde, dass es keine Vorschriften zur Dokumentation von Erfahrungen gibt, dass die Motivation und die Zeit dazu fehlt und es unklar ist welche Erfahrungen dokumentiert werden müssen und vor allem wie man Erfahrungen dokumentieren kann (Anmerkung von den Interviewten: Hier könnte eine Strukturierung und Vorlage zur Dokumentation von Erfahrungen helfen).

Fazit:

- Was die Dokumentation von wiederverwendbaren Informationen und Wissen betrifft, so beschränkt sich dies hauptsächlich auf Kundenkontakte, Produktbeschreibungen im Wiki, Ausarbeitung von Ideen im Wiki und Erläuterungen direkt im Code.
- Die Kommunikation während der SW- Entwicklung läuft größtenteils verbal oder über E-Mails.
- Information und Wissen aus der Entwicklung für die spätere Wiederverwendung werden nur in seltenen Fällen schriftlich fixiert, weil ein Großteil dieser Erfahrungen verbal kommuniziert wird. Auf diese Weise können entwicklungsrelevante Informationen schnell und effektiv ausgetauscht werden.
- In der Phase der Kodierung werden viele Information über einen Issue Tracker oder Changelogs (über subversion) ausgetauscht .
- Wenn Erfahrungen dokumentiert werden dann informell im Wiki. Das Wiki ist auch das verwendete System für das Ausarbeiten von neuen Produktideen.

5.3.2 Untersuchung bei Empolis

5.3.2.1 Dateiablage im Windows Explorer

Es wurde häufig beobachtet, dass die Mitarbeiter temporäre Verzeichnisse pflegen. Die darin enthaltenen Dateien werden regelmäßig gelöscht bzw. in vorhandene Strukturen verschoben. Diese Vorgehensweise soll sicher stellen, dass die Mitarbeiter während ihrer täglichen Arbeit nicht durch langwierige Ablageentscheidungen abgelenkt werden. Neue Dateien werden so also ad hoc in temporären Ordnern abgelegt und können sofort verwendet werden.

In der Gestaltung der nicht-temporären Struktur spiegelt sich deutlich das jeweilige Tätigkeitsprofil wieder. Software-Entwickler passen beispielsweise ihre Strukturen an die Anforderungen der Entwicklungsumgebung an und übertragen diese teilweise sogar auf andere Bereiche. Mitarbeiter, welche mit Projektmanagementaufgaben betraut sind, richten die Gestaltung der Ablagestrukturen dagegen stark auf die Anforderungen der jeweiligen Projekte aus. Im einfachsten Fall findet man ausschließlich einen Ordner „Projekte“ mit den Unterordnern der einzelnen Projekte.

Viele Mitarbeiter nutzen die Ordnerstruktur auch dazu, ihre Aufträge zu verwalten.

5.3.2.2 Ablage im Email-Client (Outlook)

Alle bei Empolis befragten Mitarbeiter nutzen Microsoft Outlook als Email-Client. Die Mitarbeiter achten in der Regel darauf, dass die Emails in ihrem Posteingang möglichst zügig abgearbeitet werden. Das bedeutet, dass abgelaufene Mails gelöscht und relevante Mails in die bestehende Struktur abgelegt werden.

Die Ablagestruktur orientiert sich - wie bei der Gestaltung der Dateiablage im Windows Explorer auch – an den jeweiligen beruflichen Aufgaben. Dies schlägt sich jedoch hauptsächlich in der Festlegung von Prioritäten für bestimmte Strukturen nieder. Im Vergleich zur Dateiablage findet man in den Email-Ablagen ein breiteres Spektrum an Inhalten. Dies ist nachvollziehbar, da die Ablage im Mailclient der vernetzten Arbeit in den Projektteams Rechnung tragen muss.

5.3.2.3 Pflege der Browser-Lesezeichen

Bei der Untersuchung der Lesezeichen in den von den Mitarbeitern genutzten Browsern zeigte sich, dass dieser Ablageort keine zentrale Rolle spielt. Fast alle Mitarbeiter haben Lesezeichen mehr oder weniger umfangreiche Sammlungen angelegt. Ein großer Teil der abgespeicherten Lesezeichen liegt jedoch brach und wird nicht mehr genutzt. Viele Lesezeichen verlieren im Laufe der Zeit ihre auch Gültigkeit und die damit verknüpften Webseiten sind überhaupt nicht

mehr erreichbar. Offenbar ist der Aufwand für viele Mitarbeitern zu hoch, ihre Browser-Lesezeichen regelmäßig zu pflegen und zu aktualisieren. Außerdem ist die Schwelle sehr niedrig, neue Lesezeichen anzulegen, welche die Gesamtmenge nur noch unüberschaubarer machen.

Einige Mitarbeiter pflegen aus diesem Grund nur eine rudimentäre Ablage mit Lesezeichen, welche nur die für die individuelle Arbeit nötigsten Lesezeichen enthält. Diese Lesezeichen sind häufig von so zentraler Bedeutung, dass sie prinzipiell für alle Mitarbeiter relevant sind.

5.4 Technische Infrastruktur

5.4.1 Befragung bei Empolis

Die Liste der Tools gilt für den Standort Kaiserslautern. Die Toolsuite ist in den jeweiligen Standorten unterschiedlich.

Allgemeine Projekt-Werkzeuge

- Das verwendete **Wiki** „MASE“ basiert auf dem JSP-WIKI und wird zum Teil zur Projektabwicklung und Projektdokumentation eingesetzt. Neben diesem allgemeinen Wiki, gibt es zumindest ein weiteres Projekt-spezifisches Wiki. Die von MASE zu JSPWiki zusätzlich implementierten Features werden wenig bis gar nicht genutzt.
- **Outlook / Exchange** wird genutzt, um emails, Kontakte und Tasks auf persönlicher und gruppenbasierter Ebene zu verwalten. Exchange wird bei einem ASP gehostet. Aufgrund der Abrechnung nach Volumen wurden die öffentlichen Verzeichnisse deaktiviert, das Wiki soll an deren Stelle treten.
- **Messenger** (ICQ, NetMeeting) sind zugelassen und werden für die interne, synchrone Kommunikation bei empolis verwendet.
- Es existiert ein **Projektverzeichnis** (Fileserver) pro Projekt wobei jeweils lokale Rechner pro Standort die entsprechenden Verzeichnisse hosten.
- Zum **Projektcontrolling** wird derzeit das Werkzeug Help4Project eingesetzt. Dieses server-basierte Tool erlaubt die Tätigkeitserfassung sowie die Zuweisung von Aufwänden und Aufgaben auf Mitarbeiter. Hersteller ist die KMU ProForSess, welche dieses Tool in erster Linie für empolis entwickelt hat.
- Weitere Wissensquellen sind:
 - WWW: google und msdn
 - Intranet: Adressen, Marketing, Produktinformation
 - Newsgroups: z.B. über Rechtsfragen

- Zeitschriften und Bücher:
- Weitere Kommunikation: Telefon und Raucherzimmer
- Als Trouble-Ticket System / Help-Desk Support wird aktuell eine Eigenentwicklung basierend auf orange insbesondere im Bereich des Supports eingesetzt.

Anforderungs-Werkzeuge

- Das **Wiki** „MASE“ wird ebenfalls zur Diskussion von Features (Anforderungen) eingesetzt.
- Zum Management von Anforderungen und Features wird teilweise der Mercury TestDirector (8.0) verwendet. Demnächst wird Telelogic Doors Unternehmensweit eingeführt.
- In den **Projektverzeichnissen** (als auch dem WIKI) werden projektspezifisch Dokumente abgelegt. Microsoft Office (insb. Microsoft Word, Power Point) wird für die Anforderungsdokumente verwendet.

Entwicklungs-Werkzeuge

- Bei der Entwicklung der Softwaresysteme werden und wurden die Werkzeuge eclipse (Java-IDE), jBuilder (Java), gcc (C++ Compiler) und MS VisualStudio (.NET/C#) eingesetzt. Der Quellcode wird teilweise mittels **Doxygen** und **JavaDoc** im Quelltext dokumentiert.
- Zur Versionierung der Softwaresysteme (Software Configuration Management, SCM) wird aktuell das Werkzeug **CVS** eingesetzt. Das Werkzeug Subversion wurde evaluiert aber **CCM** wird ab diesem Jahr einheitlich bei empolis eingeführt.

Test-Werkzeuge

- Zum Testen des Softwaresystems werden die Frameworks junit, vJUnit und cppUnit eingesetzt.
- Zum Management von Defekten wird der Mercury TestDirector (8.0) verwendet.

Dokumentations-Werkzeuge

- Die Dokumentation der Softwaresysteme für Benutzer (d.h. Die Beschreibung wie das System zu nutzen ist) wird mittels MS Word, HTML oder der Windows-Hilfe realisiert. Gespeichert werden diese Informationen im Company-Folder.

5.4.2 Befragung bei der brainbot technologies AG

Die Liste der Tools gilt für den Standort Mainz.

Allgemeine Projekt-Werkzeuge

- **Wiki** basierend auf der Programmiersprache Python namens „MoinMoin“ (seit neuestem Trac) wird teilweise zur Projektabwicklung und Projektdokumentation eingesetzt.
- Ein **email-Server** (IMAP) wird genutzt während der Projektakquise und – durchführung untereinander und mit dem Kunden zu kommunizieren. Neben Ordnern für eigene eMails die teilweise für Kollegen freigeschaltet werden können existiert noch eine eMail-Gruppe für alle Mitarbeiter. Inhalte sind Termine, Feature-Wünsche,
- Weiterhin existieren **Verzeichnisse** (*Company-Folder*) mit Literatur zu produktspezifischen Themen (z.B. Information Retrieval).

Anforderungs-Werkzeuge

- Das Python-**Wiki** „MoinMoin“ wird ebenfalls zur Diskussion von Features (Anforderungen) eingesetzt.
- In den **Verzeichnissen** (*Company-Folder*) werden projektspezifisch Dokumente abgelegt. Microsoft Office (insb. Microsoft Word, Power Point) wird für die Anforderungsdokumente verwendet. Für diese Dokumente werden neben MS-Office auch andere Softwarepakete verwendet (z.B. Adobe Photoshop).
- Zum Management von weiteren Features wird das Defect & **Issue Tracking System** „Round-up“.

Entwicklungs-Werkzeuge

- Bei der Entwicklung der Softwaresysteme werden die Werkzeuge emacs, i-Python (Python-Shell), id-utils (Identifier management), gcc (C++ Compiler) verwendet. Der Quellcode wird mittels **ePyDoc** im Quelltext dokumentiert.
- Zur Versionierung der Softwaresysteme (Software Configuration Management, SCM) wird aktuell das Werkzeug **Subversion** eingesetzt welche das bisher verwendete Werkzeug CVS abgelöst hat.

Test-Werkzeuge

- Zum Testen des Softwaresystems wird das Programm „trace“ eingesetzt.

- Zum Management von Defekten wird das Defect & **Issue Tracking System** „Round-up“ mit web und email-Interface [siehe <http://roundup.sourceforge.net/>] verwendet.

Dokumentations-Werkzeuge

- Die Dokumentation der Softwaresysteme für Benutzer (d.h. Die Beschreibung wie das System zu nutzen ist) wird mittels MS Word, HTML oder der Windows-Hilfe realisiert. Gespeichert werden diese Informationen im Company-Folder. Aufwändige bzw. sehr umfangreiche Dokumentationen werden auch im Versionierungssystem abgelegt.

6 Problemsammlung

Die Ausgangslage besteht aus Unternehmen oder Abteilungen welche selbst Software entwickeln und in von Kunden getriebenen oder eigenen Projekten ein oder mehrere Softwaresystem(e) weiterentwickeln. Mittels der Kontextermittlung wurden für die Anwendungspartner in RISE nun folgende Problemklassen identifiziert denen konkrete Probleme zugeordnet sind:

6.1 Verteilungsproblem der Information

- Wichtige Informationen zu den Projekten, Produkten, oder Kunden stehen **verteilt** in vielen verschiedenen Datenquellen. Das bedeutet, dass wertvolle Erfahrungen und Wissen welches in Projekten gewonnen wurde wird schlecht auffindbar sind und nach Projektende häufig nicht in andere bereits laufende oder neue Projekte transferiert wird. Ob durch den hohen Zeitdruck, die Verteilung an verschiedenen Standorten, zu wenig Kommunikation oder Fluktuation der Mitarbeiter – Wissen über das Produkt, die Technologie oder die Methodik geht sehr oft verloren.
- Kommentare zum Quellcode oder über Änderungen daran stehen im Versionierungssystem für die Software (z.B. CVS oder Subversion).
- Beschreibungen der Funktionen und Komponenten stehen mittels ePyDoc oder JavaDoc im Quellcode.
- Änderungen (Changelogs) werden an alle interessierten vom CVS aus verschickt.
- Informationen zu Defekte und Features stehen (Teilweise) in Defect-Tracking bzw. Trouble-Ticket Systemen.
- Konzepte und Dokumentationen über das Softwaresystem stehen teilweise auch im WIKI
- Anforderungen bzw. Konzepte zu Softwarekomponenten werden teilweise im Wiki eingetragen, weiterentwickelt und verwaltet.
- Kundeninformationen, Termine, TODO-Listen, Adressen, Fehl- und Anwesenheitslisten stehen (teilweise) im WIKI.
- Ablagesysteme für Daten und Dokumente sind u.A. Projektverzeichnisse (lokales Dateisystem), Netzlaufwerk (externes Dateisystem), eMails (Gruppen- und Personenspezifisch), Task Cards (physikalisches Blackboard), und das WIKI.

- Externe Wissensquellen die verwendet werden sind Mail- oder Newsgroups zu Technologien wie MSDN oder das „Internet“ (Erschlossen mittel Suchmaschinen wie Google)

6.2 Problem der Suche und des Findens

- Wissen und Erfahrungen über Projekte, Produkte oder Kunden, die Expertise der Mitarbeiter sowie Übersichten und Anleitungen werden oft sehr kurzfristig benötigt und unterliegen einem ständigen Wandel. Die Auffindung von Informationen und Wissens bei Bedarf ist dadurch sehr schwierig – oft ist nicht bekannt wer etwas über eine spezielle Technologie, ein Produkt oder eine Methode weiß. Verschiedene implementierte Unterstützungssysteme in KMUs sollen helfen, dieses Problem zu lösen. Dabei wird jedoch häufig deutlich, dass die abgelegten Informationen entweder nicht auffindbar oder aber verwertbar sind.
- Bei der Suche konnten spezielle Namenskonstrukte wie CamelCases in den Seitennamen nicht aufgelöst werden.
- Die Suche in den WIKIs ist unzureichend da oft nur Schlüsselwörter gesucht werden ohne Synonyme zu beachten, man findet meist nur Seiten oder wird auf Dinge hingewiesen die man bereits kennt.

6.3 Problem der Qualität & Strukturierung

- Die **Inhalte** im WIKI sind unübersichtlich und die **Strukturierung** kompliziert. Erfahrungen und Wissen welches in einem Repository von unterschiedlichen Personen an verschiedenen Orten im WIKI gespeichert wird unterliegt sehr starken Qualitätsschwankungen. Erfahrungen beziehen sich häufig auf sehr spezielle Problematiken oder setzen Hintergrundwissen voraus welches nicht in der Beschreibung der Erfahrungen enthalten ist. Nutzer können dann diese Erfahrung nur mit weiteren Informationen nutzen – die Qualität der Nutzung ist teilweise beschränkt.
- Die Umbenennung von Seiten im WIKI ist kompliziert oder teilweise unmöglich. Einmal erzeugte Seiten haben dann ggf. mehrdeutige Namen wodurch z.B. Links sehr schwer zu merken und verwenden sind.
- Die Seitenanzahl im WIKI explodiert meist nach der Einführung und jeder strukturiert für sich selbst. Dadurch wuchern die Strukturen und werden immer unübersichtlicher. Die Nutzer sind ohne feste Vorgaben oder Qualitätsmerkmale an denen man sich orientieren könnte.
- Meist fehlt eine Notifikation über neue Seiten oder Strukturelemente wodurch Nutzer die einige Tage nicht im System waren auf neue Strukturen oder Seiten stoßen. Oft werden Neuerungen auch übersehen und so Infor-

mationen später als notwendig wahrgenommen und eigene neue Seiten in falsche Strukturen eingeordnet.

6.4 Problem der Wissensrepräsentation

- Wissen und Erfahrungen zu Projekten, Produkten oder Kunden werden selbst von erfahrenen SElern unterschiedlich dokumentiert und oft nur textuell beschrieben. Die Wiederverwendung solcher Dokumente und Erfahrungen setzt aber oft voraus, dass Inhalte einheitliche Strukturen (Templates) verwenden und auch graphisch repräsentiert werden können (z.B. als MS Word Dokument das mit dem Kunden ausgetauscht werden kann).
- Für die Verwendung der Repositories oder Beschreibungswerkzeuge für Projekte und Softwaresysteme fehlen typische WYSIWYG-Funktionalitäten die bspw. in MS Word verfügbar sind.
- Weder ePyDoc noch WIKIs erlauben eine (einfache) Integration von Bildern oder Graphiken
- Der Zugriff aus textuellen Editoren (z.B. emacs, oder eine shell) auf ein WIKI ist unkomfortable / unmöglich
- Die Editierung mit dem Wiki in der WIKI-spezifischen Sprache dauert teilweise länger als mit eMail, bei Zeitdruck werden Informationen per eMail versandt und nicht im WIKI eingetragen.

7 Erste Lösungsansätze und Forschungsfragen

7.1 Handlungsgebiete

Um die Arbeit auf einen sinnvollen Kern zu fokussieren richten wir das zu entwickelnde System auf einige wenige Anwendungsfälle aus. Zusammenfassend lassen sich für unsere beiden Anwendungspartner folgende Handlungsfelder identifizieren:

- **Suchoptimierung:** Eine verbesserte Suche nach Wissen und eine verbesserte Orientierung im Repository soll die Menge an relevanten Artefakten optimieren und den Aufwand für den Zugriff auf Erfahrungen minimieren. Seiten im WIKI werden mit Metadaten – basierend auf einer SE-Ontologie – angereichert um die Suche und Orientierung zu optimieren.
- **Strukturoptimierung:** Die Optimierung der Hierarchie und Struktur von Dokumenten aus der Softwareentwicklung (Anforderungen, Defektbeschreibungen, Projektpläne, Mitarbeiterkompetenzen, Erfahrungen, etc.) soll deren Qualität erhöhen und den Benutzer optimal mit Informationen zu seiner Problematik versorgen.
- **Informationszentralisierung:** Informationen zu einem Projekt, Produkt oder Kunden sollten an einem zentralen Ort (WIKI) zusammengeführt werden, auch um eine möglichst umfassende Durchsuchbarkeit gewährleisten zu können. Ausgewählte projekt- und produktrelevante Informationen aus der Softwareentwicklung (Anforderungen, Code(änderungen), eMails, ...) werden aus existierenden Quellen extrahiert, integriert und im WIKI abgelegt.
- **Benutzungsoptimierung:** Eine Verbesserung der Benutzbarkeit (Usability) des technischen Systems soll den Benutzer zielgerichtet und in geringerer Zeit ermöglichen eine Information (z.B. passende Lösung) im WIKI zu finden oder eine neue Information (z.B. neue Anforderung) mit Hilfe ähnlicher Informationen und maschineller Unterstützung einzugeben oder zu erarbeiten.

Die primären Zielgruppen die durch das RIKI unterstützt werden sollen sind die Projektmanager, Softwareentwickler und Dokumentationsingenieure.

7.2 Nutzungsszenarien

Um die Funktionalitäten des RIKI (Reuse-oriented WIKI System) zu demonstrieren sind im folgenden einige Szenarien angegeben:

- **Wiederverwendung in der Projektakquise & -management:** Der Verantwortliche hat die Möglichkeit über eine Suche ähnlicher Projekte mit dem gleichen oder ähnlichen Kunden zu Identifizieren. Daraus kann er eine Risikoabschätzung basierend auf den Kundeninformationen oder den Projektinformationen vornehmen. Bisher gemachte Erfahrungen oder Kommentare aus der eMail-Korrespondenz kann mit in die Projektakquise einbezogen werden. Er oder der spezifische Projektmanager kann schließlich noch den Ressourcenbedarf (Mitarbeiter, Werkzeuge, ...) und eventuell betroffene Software(sub-)systeme ermitteln und eine optimale Teamstruktur basierend auf den benötigten Kompetenzen und entsprechend der Auslastung einzelner Mitarbeiter vornehmen. Die wesentlichen Fragen die das System hier beantworten können sollte sind:
 - „Wer hat schon mit diesem Kunden zu tun gehabt?“
 - „Wer kennt sich mit den benötigten Technologien aus?“
 - „Wer kennt die betroffenen Module in den In-house Produkten?“
- **Wiederverwendung in der Anforderungsbeschreibung:** Bei der Ermittlung der Anforderungen die im Projekt umgesetzt werden kann der Verantwortliche ähnliche Anforderungen suchen um diese als Vorlage zu verwenden oder um Eigenschaften wie Komplexität oder Zeiträume zur Umsetzung abzuschätzen. Weiterhin können über die Anforderungen Beziehungen zum Quellcode abgeleitet und so betroffene Subsysteme oder Experten für diese Subsystems identifiziert werden (sowie ggf. Konflikte mit anderen Projekten frühzeitig entdecken). Die wesentlichen Fragen die das System hier beantworten können sollte sind:
 - „Welche Eigenschaften haben ähnliche Anforderungen?“
 - „Warum wurden bisher die verwendeten Technologien eingesetzt?“
 - „Was wird typischerweise bei (ähnlichen) Anforderungen vergessen?“
- **Wiederverwendung beim Testen und der Entwicklung:** Der Entwickler hat die Möglichkeit über eine Suche Informationen über Experten oder Testverfahren und Erfahrungen mit Werkzeuge oder Algorithmen abzurufen und wiederzuverwenden. Die wesentlichen Fragen die das System hier beantworten können sollte sind:
 - „Wie implementiert oder testet man den Algorithmus?“
 - „Wer hat bereits an der Komponente gearbeitet und kann helfen?“
 - „Was wird typischerweise bei (ähnlichen) Anforderungen vergessen?“
- **Wiederverwendung bei der Dokumentation:** Bei der Erstellung und Pflege der Dokumentation zu einem Softwareprodukt kann der Dokumentations-Ingenieur auf bisherige Projekte zurückgreifen und ermitteln ob eine Anforderung in einem neuen Projekt die Dokumentation des Softwareproduktes beeinflussen wird. Die wesentlichen Fragen die das System hier beantworten können sollte sind:
 - „Welchen Einfluss hatten ähnliche Anforderungen auf die Doku?“
 - „Welchen Einfluss hatten ähnliche Anforderungen auf die Demos?“
 - „Wer sind die Ansprechpartner falls ein Subsystem geändert wird?“

- **Wiederverwendung von Erfahrungen:** Im Rahmen der o.g. Szenarien können neben der Wiederverwendung von ähnlichen Dokumenten auch Erfahrungen (insbesondere Handlungsanweisungen) einfließen. Durch diese Erfahrungen wird erklärt, wie man die Erstellung des jeweiligen Dokumentes zu bewerkstelligen hat. Die wesentlichen Fragen die das System hier beantworten können sollte sind:
 - Welche Erfahrungen aus diesem und anderen, ähnlichen Projekten sind für das aktuelle Dokument relevant.
 - Wie valide sind diese Anforderungen, .d.h. wie haben sich diese Anforderungen in den jeweiligen Projekten bewährt.

7.3 Lösungsansätze & Handlungsanweisungen

Um die aufgelisteten Probleme zu beseitigen oder zumindest abzuschwächen werden im Projekt verschiedene Lösungsansätze parallel verfolgt und umgesetzt. Im Projekt RISE werden wir dafür die nachfolgenden Technologien verwenden:

- **Nutzung von WIKIs:** Durch den Einsatz von WIKI's¹ soll ein verstärkter asynchroner Austausch von Projekterfahrungen unter den Mitarbeitern ange-regt und zur expliziten Dokumentation dieser Erfahrungen motiviert werden. WIKIs sollen die eigenständigen Wissensaustausch fördern und mittels einer technischen Schnittstelle (z.B. XML-RPC) die Integration weiterer Dienste fördern.
- **Wikitology:** Die Anreicherung von Artefakten mit Metadaten soll die Ver-netzung und Suche von Artefakten verbessern. Des Weiteren wird die Ver-wendung von Ontologien bei der ähnlichkeitsbasierten Suche von Artefak-ten genutzt um die Suche zu verbessern.
- **Verwendung von Technologien aus dem Semantic Web und Semantic Grid:** Durch den Einsatz des SIP (Sekt Integration Plattform) können intero-perable Dienste zur internen und externen Suche sowie zur Informationsauf-bereitung erstellt werden. Weiterhin ermöglicht das verwendete Service Grid die einfache Erstellung von komplexeren Diensten um individuelle Dienste für spezifische Inhalte von SE-Dokumenten zu erstellen.

Zur Erreichung der identifizierten Ziele und Nutzungsszenarien werden folgende Dienste entwickelt:

Dienste zur Suchoptimierung

- **Interne Quellen:** Die Suche auf internen Informationen im WIKI soll durch die Wikitologien unterstützt werden.

¹ Ein Wiki ist ein sehr einfach benutzbares, offenes Content Management System (CMS). Es handelt sich da-bei um einen Webserver mit Webseiten, die jederman verändern kann.

- Externe Quellen: Die Integration von externen öffentlichen Quellen soll den Suchraum erweitern und dem Suchenden externes Wissen bspw. aus Newsgroups² oder FAQ's³ unkompliziert zur Verfügung stellen.
- Suchmasken: Durch den Einsatz individuell anpassbarer Masken bei der Suche von Artefakten sollen die verschiedenen Ziele und Nutzungskontexte der Anwender berücksichtigt werden.

Dienste zur Benutzungsoptimierung

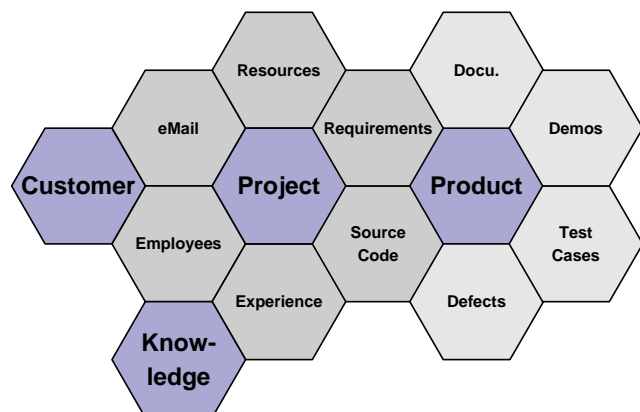
- **Benutzerprofilermittlung:** Ein Dienst welches Mitarbeiter mit Wissensbereichen verbindet um so zu jedem Mitarbeiter ein Kompetenzprofil zu ermitteln. Die Extraktion von Informationen aus verwendeten Werkzeugen und Datenquellen wie bspw. einem WIKI, CVS, oder den Chatprotokollen(?) kann dazu verwendet werden Themen und Dokumente mit Mitarbeitern zu assoziieren. Dies dient dazu Benutzerprofile anzureichern und eine Benutzer-Wissenslandkarte / Gelbe Seiten anzulegen.
- Eingabemasken: Bei der Eingabe neuer Informationen wird der Nutzer über mögliche Klassifikationen (basierend auf Wikitology) und minimale Inhaltsangaben hingewiesen („Was sollte er eingeben damit die Wiederverwendung unterstützt wird?“).
- Automatische Unterstützung der RISE Nutzer bei der Erfahrungseingabe (Richtlinien, Tips, Tags, ...)
- **Artefaktanreicherung:** Die Anreicherung von Artefakten mit Hilfe von anderen weiterführenden oder erklärenden Artefakten (Information Seeding) soll die Wiederverwendbarkeit erhöhen und die Zeit an weiterführende Artefakte zu gelangen verkürzen.
 - Automatische Unterstützung der RISE Nutzer bei der Erfahrungsnutzung (ähnliche Erfahrungen, MDA's, ...)
- **Individualstrukturen:** Jeder Nutzer verfügt über eine eigene individuelle Struktur die bei seinem Benutzerprofil gespeichert wird. Anfänglich startet er mit einer "allgemeinen" Struktur bspw. der Klassifikation des Erstellers der neuen Seite. Später können neue Seiten auch automatisch in die Individualstruktur eingeordnet werden.
 - Neben der Einordnung einer neuen Seite in der Struktur des Nutzers über die Ähnlichkeit könnte auch die Einordnung über die Einordnung anderer (d.h. so etwas wie "Collaborative Filtering/Structuring")
 - Eine Erweiterung wäre das Browsen in den Strukturen anderer um sich ggf. besser zurechtzufinden.

² Newsgroups dienen dem Austausch von Nachrichten (auch Artikel genannt), häufig über das NNTP-Protokoll.

³ Die Abkürzung FAQ bedeutet Frequently Asked Questions. Es handelt sich dabei um eine Zusammenstellung von oft gestellten Fragen und den dazugehörigen Antworten zu einem Thema.

Dienste zur Informationszentralisierung

- **Informationszentralisierung (SPOCK):** Ein Portal (Single Point of Consistent Knowledge (SPOCK)) für ein Softwareunternehmen welches Informationen aus verschiedenen allgemeinen (z.B. eMail) und SE-spezifischen Datenquellen (z.B. Subversion) extrahiert, integriert und zur zentralen Wiederverwendung ins WIKI einspeist. Da die Synchronisation technisch und inhaltlich ein größeres Problem darstellt wird zuerst nur der Informationsfluss in das Wiki mittels verschiedener Dienste betrachtet (d.h. Read-only Informationen im WIKI).
- **Informationsintegration:** Integration von Informationen rund um Projekte, Produkte und Kunden. Dies beinhaltet am Beispiel eines Projektes mindestens die Informationen zu Ressourcen, eMails, Mitarbeiter, Quellcode und Anforderungen (siehe Bild).
- eMail-Integration: ein eMail-Interface für Wikis anbieten, um diese auch in Wiki Seiten einzubinden (Wie sollen diese Mails in die Wiki Seiten eingebunden werden? (Idee für die Co-Integration bestehender Systeme)). Bspw. könnten Emails an Projektgruppen (z.B. rise_all) auch an das WIKI geschickt werden um dort (angehängt an die spezifischen Projektseiten) für alle zugänglich zu sein.
- Sourcecode-Integration um Kommentare aus CVS-Servern und dem Quellcode (als zusätzliche Information zur Kontextbestimmung) in das WIKI einzubinden (z.B. als Anreicherung der Dokumentation zu einem Softwaresystem).
- Bspw. können Autoren aus dem CVS den Komponenten zugeordnet werden oder Änderungen am Quellcode / der Funktionalität im WIKI vermerkt werden.



Methodisch müssen dazu folgende Methoden und Techniken entwickelt werden:

- **RIKI AEB-Methodik:** Entwicklung einer Methodik zum maßgeschneiderten Aufbau, Entwicklung und Betrieb des RIKIs bei neuen Anwendern. Dies beinhaltet die Ableitung von Anforderungen und Nutzungsszenarien, die Erhebung von Prozessen und der technischen Infrastruktur, der Adaption von Nutzungsänderungen durch die Community, der Erweiterungen der Integrationsplattform sowie der Information-Spaces.
- **Information Seeding:** Zu allererst werden im Rahmen dieser Methode die zum Betrieb des RIKI nötigen Inhalte initial erzeugt. Im Vorfeld liegt hier das besondere Augenmerk auf der Annotierung vorhandener Information mit Metadaten, um sie RIKI und damit den Benutzern auffindbar zu machen und eine didaktische Aufbereitung zu ermöglichen. Weiterhin identifiziert diese Methode welche Information ggf. neu zu erstellen sind. Damit wird durch diese Methodik verhindert, dass das RIKI aufgrund fehlender Inhalte nicht genutzt wird. Ein weiterer wichtiger Aspekt des Information Seeding ist Inhalte, die nicht direkt mit dem Arbeitskontext verwandt sind, dem Benutzer zu unterbreiten. Das Domänenmodell beinhaltet Relationen die genau diese Themenübergreifenden Beziehungen beschreiben. So können themennahe Inhalte, die evtl. einen neuen Wissensgebiet beinhalten, dem Benutzer hilfreich sein um seine Kompetenzen und Fähigkeiten auf andere Gebiete auszuweiten. Beispiel: Ein Anforderungsingenieur sucht nach Vorlagen und Anleitungen um Anforderungen aus Entwicklersicht zu formulieren. Durch das Information Seeding, weist RIKI den Benutzer auf die Methode „Anforderungsinspektion“ hin. Diese Inspektionsmethode kann dem Benutzer bis dato unbekannt sein - Inspektionen sind aber ein extrem wichtiges Thema während der Anforderungsphase. Ohne gebietsübergreifende semantische Relationen hätte der Benutzer das für ihn unbekanntes Themenfeld „Anforderungsinspektion“ nicht gefunden.
- **Didaktische Strukturierung:** Weiterentwicklung von Vorlagen und Ontologien zur Beschreibung, Strukturierung und Didaktisierung von Inhalten in einem RIKI.

7.4 Forschungsfragen

Allgemeine Forschungsfragen des RISE-Projektes:

- Welcher Mix aus Technologien und in einem Entwicklungsteam akzeptierte Vorgehensweisen fördert die Wiederverwendung in der Software-Entwicklung?
- Wie kann man die in einem Wiki vernetzten Informationsbausteine situationgerecht dem Benutzer zugänglich machen um die Wiederverwendung zu verbessern?

Forschungsfragen in der Pädagogik (AG Arnold):

- Welche (zusätzlichen) Informationen (Kontext) müssen beim Wissensmanagementprozess zur Verfügung gestellt werden, damit Software-Entwickler (Nutzer) neues Wissen generieren können?
- Welche unterstützenden Hilfen können den Software-Entwicklern an die Hand gegeben werden, damit diese bei der Erzeugung neuer Inhalte die Regeln und Strukturen beachten und neben dem Informations- auch Wissensmanagement methodisch gestützt wird?
- Ist es möglich, über die Wissensvermittlung hinaus auch die Kompetenzentwicklung der Anwender mittels eines solchen technischen Systems zu unterstützen? Welche unterstützenden Elemente müssen hierfür geschaffen werden (Communities of Practice, Möglichkeiten des informellen Lernens...)?

Forschungsfragen im Wissensmanagement (DFKI):

- Sind Enabler und Disabler für die Wiederverwendung bei den genannten Nutzungs- und Administrationsszenarien gleich?
- Lassen sich für die Wissensablage Strukturpatterns bestimmen, d.h. Schemata in Strukturen (von Wiki-Seiten und deren Verlinkung), ähnlich wie Programming Patterns bei der Programmierung von Software, anhand derer man die Wiki-Struktur auf einer höheren Ebene beschreiben kann?

Forschungsfragen im Software Engineering (IESE):

- Wie entwickelt man flexible Erfahrungsmanagementsysteme für agile SE-Unternehmen die sich leicht bei Änderungen der Infrastruktur oder der Prozesse anpassen lassen?
- Wie verbessert man die Verständlichkeit und Anwendbarkeit von elektronisch gespeicherten Informationen und Erfahrungen?
- Wie extrahiert man die latent vorhandenen Informationen aus SE-Datenquellen und wie beutet man diese aus um die Wiederverwendung von Informationen über Softwareprojekte und -produkte zu unterstützen?

Referenzen

- Dreher, M./Dreher, E. (1982): Gruppendiskussion. In: Huber, G./Mandl, H. (Hrsg.): Verbale Daten. Weinheim/Basel, S. 141ff.
- Merton, R. K. (1987): The Focused Interview and Focus Groups – Continuities and Discontinuities. In: Public Opinion Quarterly 51, S. 550ff.
- Morgan, D. L. (1997): Focus groups as qualitative research. Newbury Park
- König, E.; Volmer, G. (1994): Systemische Organisationsberatung: Grundlagen und Methoden. System und Organisation, Bd. 1, Weinheim, Deutscher Studien Verlag.
- Lamnek, S. (1995): Qualitative Sozialforschung. Bd. 2, Methoden und Techniken. 3. Aufl., Weinheim, Beltz, Psychologie-Verlags-Union.
- Lamnek, S. (1998): Gruppendiskussion. Theorie und Praxis. Weinheim
- Bohnsack, R. (1999): Rekonstruktive Sozialforschung. Opladen

Dokumenten Information

Titel: RISE – Reuse im Software Engineering
Evaluationsmethodik der Kontextermittlung

Datum: 01. August 2005
Report: IESE-032.05/D
Status: Final
Klassifikation: Öffentlich

© Copyright 2005, RISE Project, Gefördert durch das BMBF im Programm „SE 2006“

Alle Rechte vorbehalten.

Dieses Werk darf ohne vorherige schriftliche Erlaubnis des Herausgebers in keiner Weise, auch nicht auszugsweise, insbesondere elektronisch oder mechanisch, als Fotokopie oder als Aufnahme oder sonstwie vervielfältigt, gespeichert oder übertragen werden.

Copyright 2005, Fraunhofer IESE.

Alle Rechte vorbehalten. Diese Veröffentlichung darf für kommerzielle Zwecke ohne vorherige schriftliche Erlaubnis des Herausgebers in keiner Weise, auch nicht auszugsweise, insbesondere elektronisch oder mechanisch, als Fotokopie oder als Aufnahme oder sonstwie vervielfältigt, gespeichert oder übertragen werden. Eine schriftliche Genehmigung ist nicht erforderlich für die Vervielfältigung oder Verteilung der Veröffentlichung von bzw. an Personen zu privaten Zwecken.