Diplomarbeit
Diplomstudium Wirtschaftsinformatik FH
Business Engineer
AKAD

Wissen erhalten trotz Redimensionierungen

Praktisches und nutzenstiftendes Wissensmanagement am Beispiel der Prozessmodelle Rational Unified Process (RUP) und Extreme Programming (XP)

eingereicht von
Katja Stucki
8. Semester Wirtschaftsinformatik FH
Matrikelnummer 91703686

Florenstrasse 53, 8405 Winterthur

Korrektor (sachlicher Teil)
Hans-Peter Korn

Korrektor (formaler Teil)
Sandra Schlick

Winterthur, den 23. Oktober 2003
(Mit Anmerkungen und Modifikationen von Hans-Peter Korn)

Inhaltsverzeichnis

Inhaltsverzeichnis....I
Abbildungsverzeichnis....III
Tabellenverzeichnis....III
Abkürzungen....IV
1Grundlegung....1
1.1Problemstellung....1
1.2Gang der Untersuchung....2
1.3Grundbegriffe und –konzepte....3
1.3.1Wissen & Wissensmanagement....3
1.3.2Software, Softwareentwicklung und Prozessmodelle....5
1.3.3Konventionen....6
2Wissensmanagement im Softwareprozess....7
2.1Der Wissensmanagementprozess....7
2.1.1Überblick....7
2.1.2Die strategische Seite....9
2.1.3Die taktische Seite....10
2.2Angewandte Methode....12
2.2.1Grundlagen....12
2.2.2Ziel....12
2.2.3Bezugsrahmen....12
2.2.4Modifikation des Diagnoseinstrumentes....13
2.2.5Fragestellung....13
2.2.6Vorgehen....13
2.2.7Auswertung....14
2.3Die relevanten Wissensmanagementprozessschritte in der Softwareentwicklung....14
3Extreme Programming....18
3.1Überblick....18
3.2Die vier Variablen....20
3.3Die Grundwerte....21
3.4Die Praktiken....21
3.5Die Rollenverteilung....24
4Der Rational Unified Process....26
4.1Überblick....26
4.2Ziele....27
4.3Prinzipien....27
4.3.1Best Practices....27
4.3.24+1 View....29
4.4Prozessdefinition....30
5Beurteilung....34
5.1Vorgehen und Konventionen....34

5.2Voraussetzungen schaffen....35
5.2.1Extreme Programming....35
5.2.2Rational Unified Process....36
5.3Die taktischen Schritte....38
5.3.1Beschaffen....38
5.3.2Nutzen....40
5.3.3Lernen....42
5.3.4Einen Beitrag Leisten....44
5.4Die strategischen Schritte....47
5.4.1Beurteilen....47
5.4.2Aufbauen und Pflegen....47
5.4.3Aussondern....49
5.5Fazit....50
6Zusammenfassung....57
Quellenverzeichnis....59
Eidesstattliche Erklärung....61
Anhang
A Diagnosebogen
B Herleitung des Diagnosebogens
C Ergebnisse der Diagnose
D Barth, Steve: Defining Knowledge Management
E Martin, Robert C: Agile Processes
F Probasco, Leslee: The Ten Essentials of RUP
G What is a Lightweight Methodology?
H Code Reviews Considered Hurtful
I Elssamadisy, Amr: XP on a Large Project – A Developer's View

Abbildungsverzeichnis

Abbildung 1: Die Wissenstreppe 4
Abbildung 2: Übersicht über den Wissensmanagement-Prozess....8
Abbildung 3: Die strategische Seite des Wissensmanagement-Prozesses....10
Abbildung 4: Die taktische Seite des Wissensmanagement-Prozesses....11
Abbildung 5: Ergebnisse der Diagnose bezüglich der Relevanz der Wissensmanagement-Prozessschritte im Softwareprozess....15
Abbildung 6: Ergebnisse der Diagnose bezüglich der Relevanz der Wissensmanagement-Prozessschritte, wenn Mitarbeiter entlassen werden....15
Abbildung 7: Ergebnisse der Diagnose bezüglich der Relevanz der Wissensmanagement-Prozessschritte bei der Einarbeitung neuer Mitarbeiter....15
Abbildung 8: Medianwerte der veralngten Abdeckung der WM-Bereiche....16
Abbildung 9: Der Kostenverlauf für Änderungen....19
Abbildung 10: Ablauf eines XP-Projektes....20
Abbildung 11: Tätigkeiten während der Entwicklung mit XP....24
Abbildung 12: Die beiden Dimensionen des RUP....26
Abbildung 13: Übersicht über einen iterativen Prozess 28
Abbildung 14: 4+1 View....30
Abbildung 15: Die drei Grundelemente Worker, Activity und Artifact....31
Abbildung 16: Zentrale Artifacts in RUP und der Informationsfluss zwischen ihnen 32
Abbildung 17: Requirements Workflow....33
Abbildung 18: Balkendiagramm der Erfüllung der verlangten Abdeckung der einzelnen WM-Bereiche durch XP und RUP....52
Abbildung 19: Softwareentwicklungsprozess: verlangte und ermittelte Abdeckung der WM-Bereiche....53
Abbildung 20: Einstellungen: verlangte und ermittelte Abdeckung der WM-Bereiche....54
Abbildung 21: Entlassungen: verlangte und ermittelte Abdeckung der WM-Bereiche....54

Tabellenverzeichnis

Tabelle 1: Schema der Auswertung des Diagnosebogens....14
Tabelle 2: Zusammenstellung der Resultate der Diagnose....17
Tabelle 3: Resultate der Diagnose: Aussagen mit der Markierung "wäre schädlich"....18
Tabelle 4: Resultate der Diagnose für den Prozessschritt Beschaffen....38
Tabelle 5: Gegenüberstellung von XP und RUP für den Prozessschritt Beschaffen....39
Tabelle 6: Resultate der Gegenüberstellung für den Prozessschritt Beschaffen....40
Tabelle 7: Resultate der Diagnose für den Prozessschritt Nutzen....40
Tabelle 8: Gegenüberstellung von XP und RUP für den Prozessschritt Nutzen....41
Tabelle 9: Resultate der Gegenüberstellung für den Prozessschritt Nutzen....41

Tabelle 10: Resultate der Diagnose für den Prozessschritt Lernen.....42
Tabelle 11: Gegenüberstellung von XP und RUP für den Prozessschritt Lernen.....43
Tabelle 12: Resultate der Gegenüberstellung für den Prozessschritt Lernen.....43
Tabelle 13: Resultate der Diagnose für den Prozessschritt Einen Beitrag Leisten.....44
Tabelle 14: Gegenüberstellung von XP und RUP für den Prozessschritt Einen Beitrag Leisten. 46
Tabelle 15: Resultate der Gegenüberstellung für den Prozessschritt Einen Beitrag Leisten.…46
Tabelle 16: Resultate der Diagnose für den Prozessschritt Beurteilen.....47
Tabelle 17: Die Imperative und Herausforderungen des Prozessschrittes Beurteilen.....47
Tabelle 18: Resultate der Diagnose für den Prozessschritt Aufbauen und Pflegen.....47
Tabelle 19: Gegenüberstellung von XP und RUP für den Prozessschritt Aufbauen und Pflegen 49
Tabelle 20: Resultate der Gegenüberstellung für den Prozessschritt Aufbauen und Pflegen…49
Tabelle 21: Resultate der Diagnose für den Prozessschritt Aussondern.....49
Tabelle 22: Gegenüberstellung von XP und RUP für den Prozessschritt Aussondern.....50
Tabelle 23: Resultate der Gegenüberstellung für den Prozessschritt Aussondern.....50
Tabelle 24: Zusammenfassung der Gegenüberstellung: Ein grosses Symbol steht für 10 kleine. 51

Abkürzungen

RUP Rational Unified Process
SW Software
UML Unified Modeling Language
WM Wissensmanagement
XP Extreme Programming

1 Grundlegung

1.1 Problemstellung

Seit 1970 findet in allen wirtschaftlichen Sektoren ein struktureller Wandel statt, im Laufe dessen die Wettbewerbsfähigkeit von Unternehmen zunehmend von der optimalen Kombination von Information und Wissen abhängig wird[1]. Die sogenannten wissensbasierten Aktivposten – die Menschen mit ihren Erfahrungen, die in Datenbanken abgelegten Informationen, die Prozesse und Verfahrensweisen – haben für viele Unternehmen an strategischer Relevanz gewonnen[2]. In diesen Wissensunternehmen ist der wichtigste Produktionsfaktor unsichtbar: Es ist das Wissen in den Köpfen der Mitarbeiter. Eine der betroffenen Branchen ist diejenige der Versicherungen und Banken.

Die Banken und Versicherungen in der Schweiz durchleben in diesen Jahren eine zweite Art struktureller Veränderung[3]: Die Kapitalmärkte haben sich derart verändert, dass es schwierig geworden ist, die Zinsen, die den Kunden im Rahmen einer Versicherung der zweiten oder der dritten Säule garantiert wurden, zu erwirtschaften. Die im Vergleich zu früheren Jahren schlechte Lage auf den Kapitalmärkten kann nur ausbalanciert werden, wenn die internen Kosten gesenkt werden. Kosten zu senken bedeutet meist auch, Stellen abzubauen.

Banken und Versicherungen verfügen häufig über eine verhältnismässig grosse interne Informatikabteilung, die unter anderem auch massgeschneiderte Softwarelösungen herstellt[4]. Solche Informatikanwendungen werden bereits seit den 60er Jahren entwickelt. Häufig handelt es sich um grossrechnerbasierte Systeme, seit ungefähr 10 Jahren kommen aber auch Client-Server-Systeme und heute vermehrt webbasierte, objektorientierte Technologien zum Einsatz.

Diese internen Informatikabteilungen sind häufig besonders massiv von Kostenreduktionsmassnahmen betroffen, im Rahmen derer Personal entlassen wird[5]. Der dadurch entstehende Verlust an Know-how-Kapital scheint unterschätzt zu werden, denn Know-how-Risiken gehören eher zu den alltäglichen Gefahren: Ereignisse wie der Weggang von Mitarbeitern oder Restrukturierungen begleiten seit jeher den Prozess der betrieblichen Zielerreichung. Die zunehmende Wichtigkeit des Know-hows als eine zentrale Ressource wissensintensiver Unternehmen würde allerdings eine erhöhte Aufmerksamkeit des Managements verdienen[6]:

"Tatsächlich beklagen heute viele Unternehmen, dass sie im Zuge von Reorganisationen einen Teil ihres Gedächtnisses verloren haben. Diese kollektive Amnesie beruht auf der

---

[1] North 2002, 15 identifiziert die Triebkräfte, die für die gesteigerte Bedeutung der Ressource Wissen wirken: Struktureller Wandel der Gesellschaft, Globalisierung und Entwicklung in der Informations- und Kommunikationstechnologie.

[2] vgl. dazu Nonaka 1997, 17, 57f, Pawlowsky 1998, 11f, North 2002, 19, 31f, Willke 2001, 26f, 289.

[3] vgl. dazu Pfiffner 2002, Gygi 2002, Rasonyi 2001.

[4] Folgende Zahlen liegen vor:

- UBS: ca. 4000 Personen sind mit der Entwicklung von Informatiklösungen betraut. (Auskunft per E-Mail durch Dr. Thomas Schneider, eBanking Solutions, UBS AG, vom 3. Juni 2003.)
- Winterthur Versicherungen: 650 Personen beschäftigen sich mit der Entwicklung und dem Betrieb der internen Informatiklösungen. (Mündliche Auskunft Martin Frick, Leiter Informatik Winterthur Versicherungen, anlässlich einer Informationsveranstaltung für die IT-Mitarbeiter der Winterthur Versicherungen vom 17. März 2003).
- Clariden betreibt keine Entwicklung von Bankensoftware. Diese wird eingekauft. Ca. 20 Mitarbeiter sind für die Entwicklung von Schnittstellen und das Erstellen von Reports angestellt. Allerdings wird die klare Unterscheidung zwischen Entwicklung (Lieferant) und Customizing, Parametrisierung (eigen Leistung) immer schwieriger. (Auskunft per E-Mail durch P. Morf, IT-Services Clariden Bank, vom 5. Juni 2003.)
- Vontobel: ca. 25 Personen sind mit der Entwicklung von Eigenlösungen beschäftigt. Ca. 80% der eingesetzten Informatikmittel sind Eigenentwicklungen. (Auskunft per E-Mail durch Walter Benz, Bank Vontobel AG, vom 3. Juni 2003.)

Auch Friedewald et al. 2001 zählen die Finanzdienstleistungsbetriebe zu der sekundären Softwarebranche, da sie einen besonders hohen Anteil an Softwareentwicklung haben. (Friedewald et al. 2001, 81)

[5] vgl. Meyer 2003.

[6] vgl. Probst et al. 1998, 233. Willke 2001, 290: Die Vernachlässigung ist auch in Wissenschaft und Politik zu beobachten. Probst et al. 1998, 45, geben einen Überblick über weitere Know-how-Risiken von wissensintensiven Unternehmen.

unbedachten Zerstörung informeller Netzwerke, welche wichtige aber wenig beachtete Prozesse steuern."[7]

Informatiklösungen werden in sogenannten Softwareprozessen nach bestimmten Vorgehensmodellen entwickelt. Im objektorientierten Umfeld kommen zur Zeit vor allem der Rational Unified Process (im Folgenden mit RUP abgekürzt) und Extreme Programming (im Folgenden im XP abgekürzt) zum Einsatz. Diese Arbeit setzt sich zum Ziel zu untersuchen, inwieweit diese beiden Prozessmodelle geeignet sind, der besprochenen kollektiven Amnesie entgegenzusteuern. Dabei soll nicht nur ihr Beitrag zur Verhinderung von organisationalem Vergessen bei Entlassungen untersucht werden, sondern auch, wie sie Wissensmanagement im Rahmen der täglichen Arbeit unterstützen. Als dritter Untersuchungsgegenstand gilt die Einarbeitung neuer Mitarbeiter: Auch hier sollte das Prozessmodell unterstützend wirken und die schnelle Einarbeitung – die schnelle Wissensentwicklung - einer neu dazugekommenen Person fördern.

Zusammenfassend erhalten wir also folgende Fragestellung:

1. Wie unterstützen die Prozessmodelle RUP und XP Wissensmanagement?
2. Wie tragen sie dazu bei, bei Entlassungen das Ausmass organisationalen Vergessens möglichst gering zu halten?
3. Wie tragen sie dazu bei, dass neu eintretende Mitarbeiter möglichst schnell zu produktiven (Wissens-)Mitarbeitern werden?

Die Beantwortung dieser drei Fragen wird es erlauben, einige Empfehlungen zum Einsatz von RUP und XP abzugeben, so dass den Anforderungen des Wissensmanagements durch die Art, wie Software entwickelt wird, bewusster entsprochen werden kann.

Die Integration der Wissensmanagementprozesse in die Softwareentwicklung ist deshalb sinnvoll, weil Softwareentwickler „Kopfarbeiter“ sind. Die Gefahr ist in diesem Umfeld besonders gross, dass Wissen in privaten Reservoirs versickert. Gerade im dynamischen Umfeld der Informationstechnologie ist es aber wichtig, Wissen zu teilen, denn nur so kann Wissen wachsen und erneuert werden. Nicht nur sollen neue Ideen entstehen und weiter geteilt werden, bestehendes Wissen soll auch so bewirtschaftet werden, dass der Verlust Einzelner nicht bedeutet, dass Wissen über ganze Teile von Applikationen verloren geht. Der Fokus des Wissensmanagements auf die bestehenden Softwareprozesse soll es ermöglichen, das Unternehmen flexibel zu halten und trotz der mit dieser Flexibilität oft einhergehenden personellen Schwankungen die Stabilität der Software zu sichern.

1.2 Gang der Untersuchung

Um die dargestellte Fragestellung untersuchen zu können, wird zuerst ein gemeinsames Verständnis der wesentlichen Begriffe "Wissen", "Wissensmanagement" und "Softwareentwicklungsprozess" erarbeitet. Anschliessend wird der Versuch der Systematisierung des Wissensmanagementprozesses von Bukowitz et al.[8] dargestellt, anhand dessen die beiden zu untersuchenden Prozessmodelle beurteilt werden können. Diese Systematisierung stellt ein Diagnoseinstrument zur Verfügung, das es erlaubt, diejenigen Aktivitäten des Wissensmanagements zu identifizieren, die für eine bestimmte unternehmerische Tätigkeit besonders wichtig sind. Das Diagnoseinstrument wird in dieser Arbeit in leicht veränderter Form auf den Softwareprozess angewendet.

Die Diagnose wird von einer Gruppe von 10 Softwareentwicklern bearbeitet, die in den beiden Finanzinstituten Winterthur Versicherungen und UBS arbeiten. Die Auswertung der Diagnose wird ein grobes Bild der im Softwareprozess kritischen Faktoren des Wissensmanagements zeichnen. Aufgrund der nicht repräsentativen Grundgesamtheit der

[7] Probst et al. 1999, 294.

[8] vgl. Bukowitz et al. 2002.

Befragten kann die Auswertung nur Tendenzen zeigen. Wesentliche Aussagen werden aufgrund der sichtbaren Tendenzen aber dennoch möglich sein.

Nachdem mit Hilfe des Diagnoseinstrumentes die Elemente des Wissensmanagements identifiziert wurden, die mit erhöhter Aufmerksamkeit zu betrachten sind, werden die beiden Softwareprozesse beschrieben. So wird ein gemeinsames Verständnis für die Charakteristika der Prozessmodelle erarbeitet. Anhand dieser Grundlegung und den Resultaten der Diagnose wird dann eine Beurteilung der Prozessmodelle vorgenommen. Diese Beurteilung wird aufzeigen, wie mit dem RUP und XP die Herausforderungen des Wissensmanagements bewältigt werden können. Abschliessend können Empfehlungen zum Einsatz der beiden Prozessmodelle abgegeben werden.

1.3 Grundbegriffe und –konzepte

1.3.1 Wissen & Wissensmanagement

Die Ansichten darüber, was Wissen ist oder sein sollte, differieren stark[9]. Die Uneinigkeit gründet in der Streitfrage, ob Wissen als Ergebnis, Resultat und Produkt existiere, oder ob es nur durch Sinneserfahrung gewonnen werden kann. Ersteres ist die Ansicht des Rationalismus: Es gibt apriorisches Wissen, das durch logisches Denken deduktiv erlangt werden kann. Die zweite Auffassung ist die Überzeugung des Empirismus: Wissen wird induktiv von bestimmten Sinneswahrnehmungen abgeleitet.[10]

Wird Wissen als Ergebnis, als Zustand oder als Produkt verstanden, so stellt es die geprüfte Erfahrung aus der Vergangenheit dar, die, in eine bestimmte Form und Struktur gebracht, überliefert wurde. "Die zugrundeliegende Metapher ist die eines Wissensgebäudes mit festen Fundamenten, an dem jede Generation weiterbaut, nachdem sie die Grundlagen erworben hat."[11] Das handelnde Individuum kann sich die in der Vergangenheit von anderen gemachten Erfahrungen aneignen und wird so zum vollkommen informierten Entscheider.

Dieses Verständnis greift aus dem Grund zu kurz, dass Wissen mit Information oder Daten gleichgesetzt wird. Führen wir, wie in den folgenden Abschnitten gezeigt, eine Unterscheidung der Begriffe Daten, Information und Wissen ein, so kann die Komplexität des Wissenstransfers und des menschlichen Zusammenlebens berücksichtigt werden.

Auch dem zweiten Verständnis von Wissen liegt die Annahme zugrunde, dass in zweckgerichtetem Handeln Wissen angewandt wird[12]. Es geht aber davon aus, dass die intentionalen, auf angewandtem Wissen beruhenden Aspekte von Handlungen durch den Handelnden selbst nur beschränkt fassbar und auf Befragung hin nicht vollständig diskursiv erklärbar sind. Es wird also unterschieden zwischen einem diskursiven Bewusstsein, dessen Wissensinhalte in Worten fassbar ist, und einem praktischen Bewusstsein, dessen Inhalte nicht oder nur schwer in Worte zu fassen sind[13]. In der Terminologie des Wissensmanagements ist das diskursive Bewusstsein der Sitz des expliziten Wissens, das praktische Bewusstsein derjenige des impliziten Wissens[14].

Die Grenze zwischen den beiden Bewusstseinsebenen ist gleitend, durchlässig und in ständiger Veränderung. Der Handelnde befindet sich in einem kontinuierlichen Lernprozess, weil der Kontext seines Handelns sich stetig verändert – nicht nur aufgrund äusserer Faktoren, sondern massgeblich auch durch sein eigenes Handeln. Jede Handlung findet also in einem Kontext statt, der durch eben diese Handlung auch verändert wird. So müssen die Erfahrungen

[9] vgl. Barth 2002: Dort stellt man gar fest, dass die wirklichen Experten den Begriff nicht verwenden, da er nicht definierbar ist und nichts mit Management im engeren Sinne zu tun hätte. Diese Sichtweise wird in dieser Arbeit nicht übernommen. Der Begriff wird verwendet gemäss der noch folgenden Definition.

[10] Die Polaritäten sind in Schneider 2001, 33f und Nonaka et al. 1997, 33f dargestellt. Nonaka et al. 1997 33 – 40 beschreiben die Entwicklung des Wissensbegriffes anhand der Schriften bedeutender westlicher Philosophen.

[11] Schneider 2001, 33.

[12] vgl. Nonaka et al. 1997, 40 und 70, 71.

[13] vgl. Giddens 1992, 52-55.

[14] vgl. Bukowitz et al. 2002, 13f.

und die zur Verfügung stehenden Informationen stetig hinterfragt und neu kombiniert werden. In diesem Prozess findet Lernen statt: neues Wissen wird entwickelt.

Informationen bilden somit eine der Grundlagen für Wissen. Der begrifflichen Trennung zwischen Information, Daten und Wissen ist grosses Gewicht beizumessen[15]. Daten wie z.B. Wertpapierkurse, der Wert des Aktienindexes, Zinssätze u.ä. sind kontextunabhängig vorliegende Zahlen und Zeichen. Daten werden zu Informationen verdichtet, indem sie in einen sinnvollen Kontext und Zweckbezug gebracht werden.

Werden Informationen zur Erkennung von Ursache-Wirkungs-Zusammenhängen herangezogen und bewertet, so entsteht Wissen. Wissen bedeutet also, in einem Kontext Daten und Informationen mit Hilfe des Intellektes vernetzen[16]: "Wissen ist mithin massgeblich das Ergebnis der Verarbeitung von Daten und Informationen durch Intelligenz und Lernen."[17] Daher ist Wissen immer an Personen gebunden, während Informationen und Daten unabhängig von Personen gespeichert werden können.

Zur Illustration der Zusammenhänge zwischen Daten, Informationen, Wissen und Handeln soll die Wissenstreppe dienen, wie sie in Abbildung 1 wiedergegeben ist.

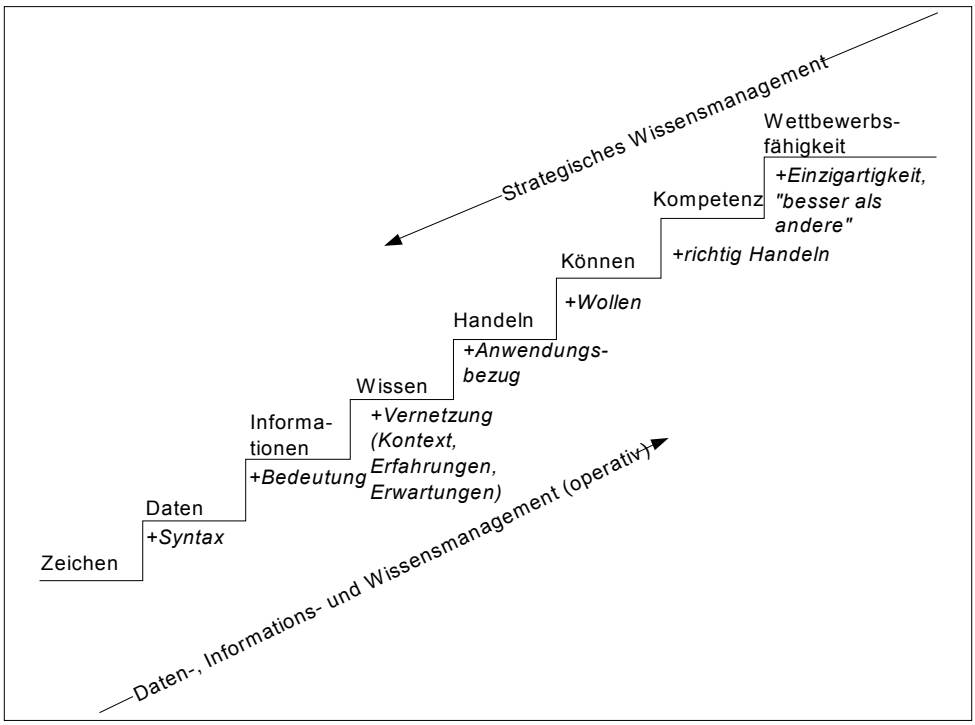


Abbildung 1: Die Wissenstreppe[18]

Die Wissenstreppe geht über die Veranschaulichung des Zusammenhanges zwischen den genannten Begriffen hinaus, um zu zeigen, wie Wissensmanagement auf die Wettbewerbsfähigkeit zu wirken vermag..

Nicht nur um seine Wettbewerbsfähigkeit zu steigern, sondern auch um den Status Quo zu bewahren, muss daher das wissensintensive Unternehmen vorhandenes Wissen optimal nutzen, weiterentwickeln und in neue Produkte, Prozesse und Geschäftsfelder umsetzen[19]. Die beiden Pfeile ober- und unterhalb der Wissenstreppe weisen darauf hin, dass diese Aufgabe aus zwei Richtungen bewältigt werden muss: Einerseits ist das Management aufgefordert, sich auf strategischer Ebene darüber Gedanken zu machen, welches Wissen für die

[15] Diese Ansicht wird u.a. geteilt mit Albrecht 2003, 191 und Willke 2001, 7-12. Die Definitionen folgen Probst et al. 1998, 22. Für weitere Begriffsdefinitionen von Daten, Informationen und Wissen vgl. Von Felbert 1998, 122, North 2002, 39 und Probst et al. 1999, 36-39.

[16] vgl. auch North 2002, 38f und Nonaka et al. 1997, 70.

[17] Von Felbert 1998, 122.

[18] nach North 2002, 39.

[19] vgl. North 2002, 3.

Wettbewerbsfähigkeit relevant ist, welche organisationalen Strukturen und welche Formen der Zusammenarbeit die Wissensbildung fördern und welche Kultur dazu notwendig ist. Andererseits müssen von operativer Seite die Daten und Informationen zur Verfügung gestellt werden, welche die Wissensbildung erlauben und fördern. Statt vom operativen wird auch vom taktischen Wissensmanagement gesprochen[20]. Damit sind diejenigen Aktivitäten gemeint, die den täglichen Einsatz von Wissen lenken und ermöglichen.

Da es kein Wissen losgelöst von Personen gibt, muss sich die Unternehmensführung über den Umgang mit Wissensträgern Gedanken machen[21]. So wird deutlich, dass sich Wissensmanagement zwar mit dem Speichern, Ordnen und zur Verfügung stellen von Informationen befasst, dass das Hauptaugenmerk aber auf die Formen der Zusammenarbeit gerichtet werden sollte[22].

Zusammenfassend sei festgehalten, was in dieser Arbeit unter Wissen und Wissensmanagement verstanden wird:

Wissen[23]

Wissen ist die Gesamtheit der Kenntnisse und Fähigkeiten, die Personen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen. Wissen stützt sich auf Daten und Informationen, ist im Gegesatz zu diesen jedoch immer an Personen gebunden. Wissen entsteht als individueller Prozess in einem spezifischen Kontext und manifestiert sich in Handlungen.

Wissensmanagement[24]

Wissensmanagement hat zum Ziel, vorhandenes Wissen optimal zu nutzen, weiterzuentwickeln und in neue Produkte, Prozesse und Geschäftsfelder umzusetzen. Es ist der Prozess, mit dem eine Organisation ihre geistes- bzw. wissensbasierten Aktiva gewinnbringend, das heisst Nutzen für den Kunden schaffend, einsetzt.

1.3.2 Software, Softwareentwicklung und Prozessmodelle

Unter Software ist die "Gesamtheit aller Programme, die auf einer Rechenanlage eingesetzt werden können"[25] zu verstehen. Dabei ist zu unterscheiden zwischen Systemsoftware und Anwendungssoftware. Die Systemsoftware umfasst alle Programme, die für den korrekten Ablauf einer Rechenanlage erforderlich sind, sowie diejenigen Programme, welche die Programmerstellung unterstützten und allgemeine Dienstleistungen bereitstellen. Unter Anwendungssoftware werden all diejenigen Programme zusammengefasst, die der Lösung von Benutzerproblemen dienen. Die vorliegende Arbeit beschäftigt sich mit dem Prozess der Erstellung von Anwendungssoftware.

Die Disziplin, welche sich damit beschäftigt, Software herzustellen, heisst Software Engineering oder Softwaretechnik. Sie wird definiert als "zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmässige Entwicklung und Anwendung von umfangreichen Softwaresystemen. Zielorientiert bedeutet die Berücksichtigung z.B. von Kosten, Zeit, Qualität."[26]

---

[20] vgl. Bukowitz et al. 2002, 20f. Dieser Ansatz wird in Kapitel 2.1 verfolgt.

[21] vgl. North 2002, 124.

[22] Worum sich Wissensmanagement dreht, ist, genauso wie der Begriff "Wissen" selbst, umstritten. Die unterschiedlichen Ansätze sind dargestellt bspw. in Lehner 2000, 225-248 und North 2002, 179f. Die hier vertretene Ansicht wird geteilt mit Von Felbert 1998, 122. Die sich daraus ergebenden Vorteile sind zusammengefasst in North 2002, 182.

[23] Die folgende Definition folgt North 2002, 39f.

[24] Die folgende Definition folgt North 2002, 3 und Bukowitz et al. 2002, 10f.

[25] Duden 1993, 665.

[26] Balzert 1998, 36.

Die Softwaretechnik hat unterschiedliche Vorgehensmodelle oder Prozessmodelle hervorgebracht.

> **Prozessmodelle** sind vereinfachende, abstrahierende Beschreibungen eines Softwareprozesses.[27]

Prozessmodelle wurden entwickelt, um die folgenden Aufgaben geordnet anzugehen[28]:

- Eine funktionierende Kommunikation im Projekt sicherstellen.
- Sicherstellen, dass jeder Projektmitarbeiter weiss, wofür er verantwortlich ist.
- Sicherstellen, dass jeder Projektmitarbeiter weiss, welchen Input er von wem bekommt und an wen er welchen Output zu liefern hat.
- Beobachten des laufenden Projektes hinsichtlich seines Fortschrittes.
- Rechtzeitiges Erkennen von Gefahren und Risiken und deren Eindämmung oder Beseitigung.

Jeder Softwareprozess stellt eine konkrete Lösungsvariante dar, mit deren Hilfe diese Aufgaben bewältigt werden können. Es ist die Art und Weise, in der Software in Firmen entwickelt wird[29]. Ein Softwareprozess wie folgt definiert:

> "A **software process** is a set of activities and associated results which produce a software product. ... There are four fundamental process activities (...) which are common to all software processes. These activities are:
> - Software specification
>   The functionality of the software and constraints on its operation must be defined.
> - Software development
>   The software to meet the specification must be produced.
> - Software validation
>   The software must be validated to ensure that it does what the customer wants.
> - Software evolution
>   The software must evolve to meet changing customer needs."[30]

Die beiden in dieser Arbeit zur Diskussion stehenden Vorgehensmodelle RUP und XP verkörpern zwei der möglichen Lösungsvarianten.

1.3.3 Konventionen

Diese Arbeit hat die Untersuchung von Vorgehensmodellen der Softwaretechnik zum Ziel, deren Begründer vor allem im englischen Sprachraum aktiv sind, in Englisch publiziert haben und deren Internetauftritte in englischer Sprache gehalten sind. Obwohl ein Teil der Publikationen auch in Deutsch vorliegt, werden in dieser Arbeit die englischen Begriffe verwendet, um die Terminologie eindeutig zu halten. Die entsprechenden Begriffe sind bei ihrer Verwendung in der `Courier-Schrift` hervorgehoben. Dieselbe Konvention wird auch auf englische Begriffe des Wissensmanagements angewendet.

[27] vgl. Sommerville 2001, 8.

[28] gemäss Versteegen 2000, 11.

[29] Einen Überblick über die gängigen Prozessmodelle bietet Sommerville 2001, 9 und 45 – 55.

[30] Sommerville 2001, 8.

2 Wissensmanagement im Softwareprozess

2.1 Der Wissensmanagementprozess

2.1.1 Überblick

"Wissensmanagement ist der Prozess, mit dem eine Organisation ihre geistigen beziehungsweise wissensbasierten Aktiva gewinnbringend ausnutzt."[31]

Eine gewinnbringende Nutzung könnte beispielweise sein, innovative Lösungen hervorzubringen, welche dem Kunden bedeutenden Mehrnutzen bringen. In innovativen Unternehmen wird auf dem vorhandenen Wissensschatz aufgebaut, so dass Wissen weiterentwickelt wird und neues Wissen entsteht. Der Grad der Innovativität hängt dabei von der Kreativität und der Problemlösefähigkeit der Mitarbeiter ab[32]. Diese beiden Fähigkeiten können kaum direkt beeinflusst werden. Indirekt vermag aber eine entsprechende Unternehmenskultur – ein so genannter `enabling context`[33] – sehr positiv zu wirken. Optimalerweise zeichnet sich diese durch die folgenden vier Merkmale aus[34]:

1. Autonomie: Die Mitarbeiter können sich autonom bewegen.
2. Intention: Die Mitarbeiter sind sich der Ziele ihres Tuns bewusst.
3. Fluktuation und kreatives Chaos: Die Arbeit findet in einem gewissen kreativen Chaos statt, wo die Fluktuation den Geist wach hält.
4. Redundanz und notwendige Vielfalt: Informationen werden auch redundant, das heisst mit keinem konkreten Anwendungsfall verbunden, weitergegeben. So fliessen sie in einer Vielfalt, welche die offensichtliche Notwendigkeit der täglichen Arbeit übersteigt.

Neben diesen vier Umweltbedingungen kann eine weitere identifiziert werden, welche die Wissensentwicklung der Mitarbeiter fördert: Jeder muss die Freiheit haben, seine Arbeit in bestmöglicher Qualität zu verrichten: "Wir neigen dazu, unsere Selbstachtung eng mit der Qualität der Produkte, die wir herstellen, zu koppeln."[35] Für die gewinnbringende Nutzung des Wissens wirkt dies synergetisch, denn Qualitätssteigerung und Produktivitätssteigerung gehen Hand in Hand. Allerdings wirkt die Qualitätssteigerung nur verzögert auf die Steigerung der Produktivität, so dass investiert werden muss – meist in Form von Zeit –, ohne dass ein unmittelbarer Mehrwert entsteht: "Qualität ist gratis, aber nur für diejenigen, die bereit sind, dafür zu zahlen."[36]

Im Versuch, die für das Wissensmanagement relevanten Faktoren zu erfassen, wurden unterschiedliche Systematisierungsansätze entwickelt[37]. In dieser Arbeit wird der Ansatz von Bukowitz et al. 2002 gewählt. Es handelt sich um einen Versuch der Gesamtkonzeption für die Bewirtschaftung der Ressource Wissen unter betriebswirtschaftlichen Gesichtspunkten.

Als Ausgangslage dient diesem Ansatz die Bildung von zwei Teilprozessen, die durch die konzeptuelle Trennung zwischen taktischem und strategischem Wissensmanagement entstehen (vgl. Abbildung 2):

- Taktisches Wissensmanagement findet mit dem täglichen Einsatz von Wissen statt, wenn beispielsweise auf Aufforderungen vom Markt oder auf günstige Gelegenheiten im Markt reagiert wird. Die Aktivitäten lassen sich in die Bereiche Nutzen, Beschaffen, Lernen und Einen Beitrag Leisten gruppieren.

[31] Bukowitz et al. 2002, 10.

[32] vgl. Pawlowsky 1998, 11f.

[33] vgl. Von Krogh et al. 2000, 49-54.

[34] vgl. Nonaka et al. 1997, 88-99.

[35] De Marco et al. 1998, 23.

[36] De Marco et al. 1998, 26f.

[37] Eine Übersicht über einige ausgewählte Wissensmanagement-Konzepte ist in North 2002, 187-203 zu finden.

- Strategisches Wissensmanagement beinhaltet die Aktivitäten, die zum Ziel haben, das geistige Vermögen eines Unternehmens den strategischen Anforderungen anzupassen. Hier lassen sich die Bereiche Beurteilen, Aufbauen und Pflegen und Aussondern unterscheiden.

Pro Wissensmanagement-Bereich können nun Imperative ausgesprochen werden, auf die das Unternehmen antworten muss, will es seine Effektivität im entsprechenden Wissensmanagement-Bereich verbessern. Innerhalb der Imperative identifizieren die Autoren Herausforderungen, denen sich das Unternehmen stellen muss.

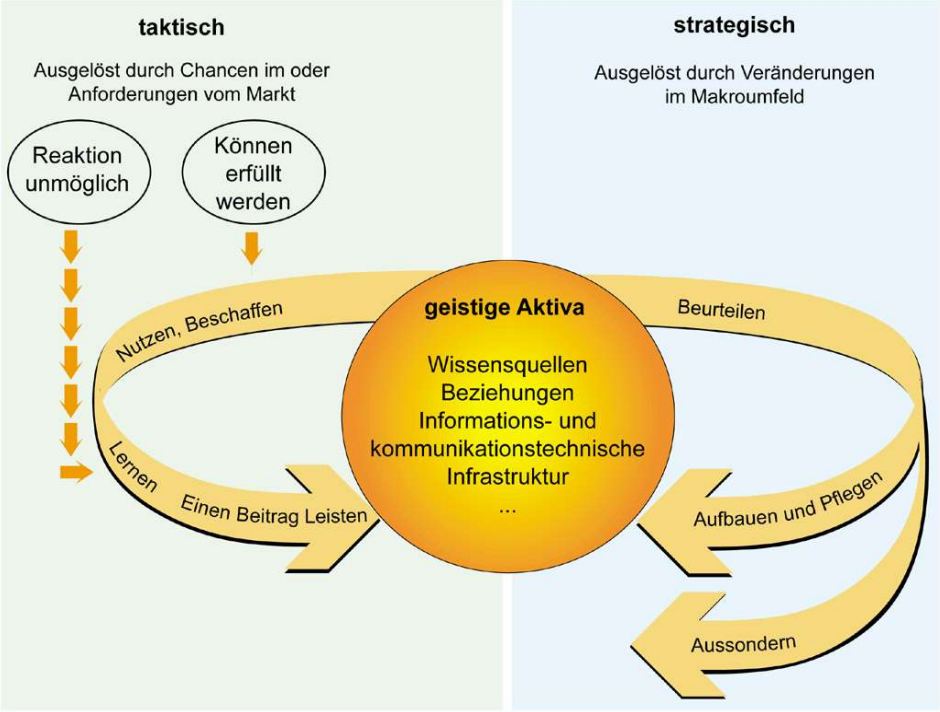


Abbildung 2: Übersicht über den Wissensmanagement-Prozess[38]

Um für ein Unternehmen oder einen Unternehmensbereich diejenigen Bereiche im Wissensmanagement zu identifizieren, die noch besonders schwach ausgebildet sind und die daher mit Priorität angegangen werden sollten, bieten die Autoren ein Diagnoseinstrument an[39].

Das Diagnoseinstrument wird in dieser Arbeit in abgeänderter Form verwendet. Es soll helfen, diejenigen Bereiche des Wissensmanagements zu identifizieren, auf die im Softwareprozess besonderes Augenmerk gelegt werden sollte, wenn man Wissen in der täglichen Arbeit optimal nutzen und den Wissensverlust bei Redimensionierungen möglichst gering halten möchte.

In den folgenden Kapiteln werden die Ziele jedes Teilprozesses erläutert. Die Imperative der einzelnen Prozessschritte werden mit ihren Herausforderungen in einer Übersicht dargestellt. Anschliessend wird das verwendete Diagnoseinstrument vorgestellt, um dann die Resultate der Diagnose darstellen zu können. In der Folge wird ein Bild der zu untersuchenden Softwareprozesse gezeichnet, so dass es möglich sein wird zu beurteilen, ob sie sich zur Bewältigung der als wichtig eingestuften Herausforderungen des Wissensmanagements eignen.

[38] nach Bukowitz et al. 2002, 21.

[39] auch Probst et al. 1998 empfehlen, als Erstes innerhalb des Unternehmens festzustellen, welche Bereiche des Wissensmanagements auf welchem Niveau bereits verwirklicht sind. Sie nennen das Resultat das "Wissensprofil" des Unternehmens (vgl. Probst et al. 1998, 236 / 389); dasselbe Vorgehen schlägt North 2002, 281 vor: Er beginnt mit der Kurzdiagnose "Wissen und Lernen" und erstellt ebenfalls ein Wissensprofil.

2.1.2 Die strategische Seite

Abbildung 3 bietet eine Übersicht über die strategische Seite des Wissensmanagement-Prozesses. Die Prozessschritte sind in den schwarz umrandeten Kästchen zu sehen. In den nicht umrandeten Kästchen sind die Imperative (fett) und Herausforderungen (als Aufzählung) jedes Prozessschrittes enthalten.

Der strategische Teilprozess hat zum Ziel, die Strategie des Wissensmanagements mit der übergreifenden Unternehmensstrategie in Einklang zu bringen. Um dieses Ziel zu erreichen, müssen die vorhandenen geistigen Aktiva laufend beurteilt und mit dem künftigen Bedarf abgeglichen werden. Abgleichen bedeutet entweder Wissen aufbauen und pflegen, oder Wissen, das nicht benötigt wird, aussondern.

Bezüglich Informationen und Daten, die in elektronischer Form oder auf Papier vorliegen, ist dieser Kreislauf weniger heikel, als in Bezug auf das Wissen, das an Menschen gebunden ist. Hier nämlich bedeutet der Wandel zum Wissensunternehmen für die Mitarbeiter und die Führungskräfte, ihre Arbeitsweisen und Rollen zu verändern[40]. Es sind neue Arten des Managements sowie neuartige Verträge zwischen Unternehmen und Mitarbeitern notwendig. Die Wissensträger müssen ihre Intelligenz, ihre Lernfähigkeit und ihr Wissen dem Unternehmen zur Verfügung stellen. Im Gegenzug dazu verpflichtet sich das Unternehmen, die Fähigkeiten und Fertigkeiten der Mitarbeiter zu nutzen, zu entwickeln und zu sichern[41]. Mit der Antwort auf die Imperative der strategischen Seite wird die Art und Weise, wie diese Pflicht erfüllt wird, mit der Gesamtstrategie in Einklang gebracht.

[40] zu diesem Schluss kommen auch die Verfechter des Konzeptes des organisationalen Lernens (vgl. Senge 2001).

[41] vgl. North 2002, 124-127.

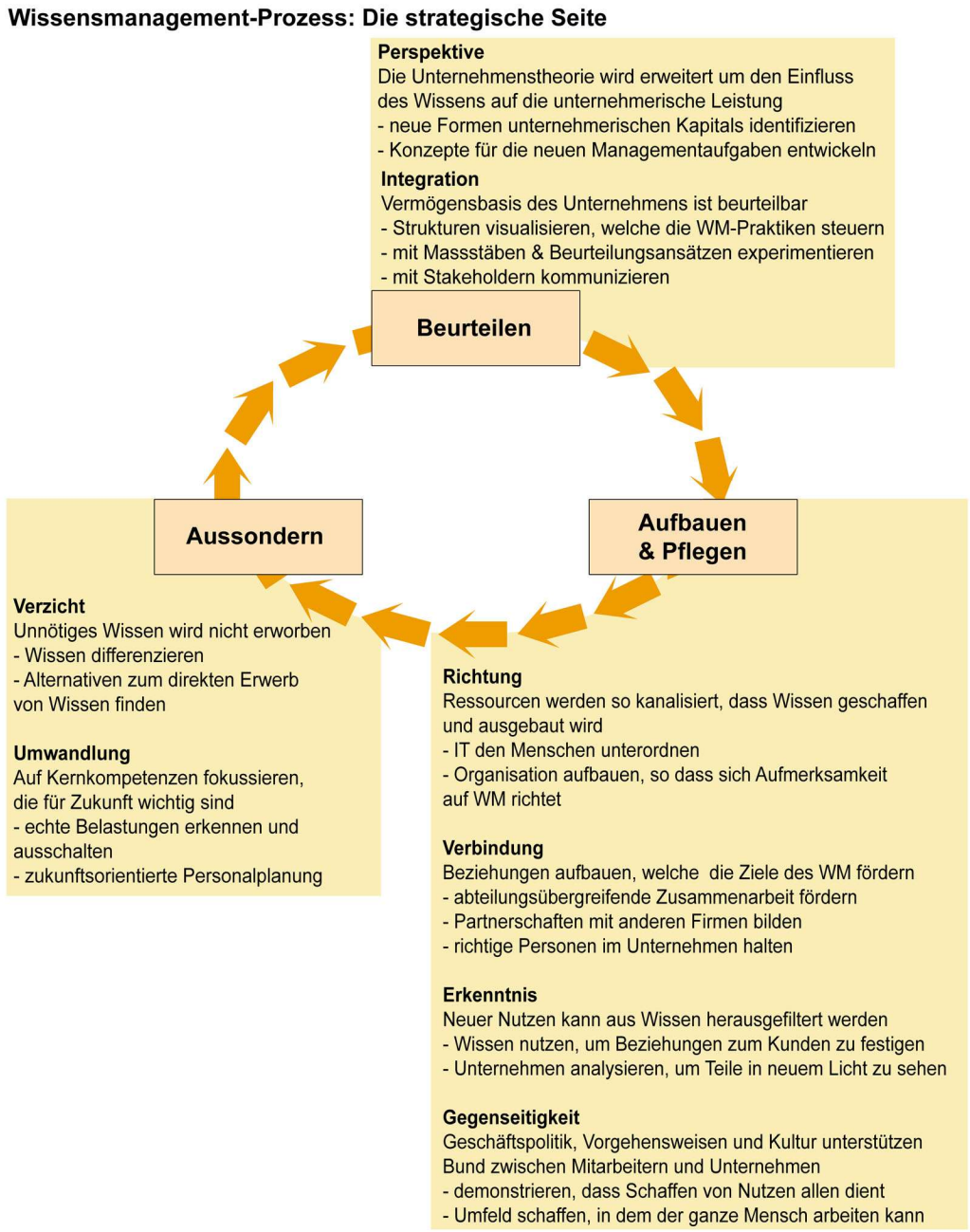


Abbildung 3: Die strategische Seite des Wissensmanagement-Prozesses[42]

2.1.3 Die taktische Seite

Zur taktischen Seite des Wissensmanagement-Prozesses gehören alle Wissensmanagement-Aktivitäten, die sich in der täglichen Arbeit niederschlagen. Sie lassen sich in die vier Schritte Beschaffen, Nutzen, Lernen und Einen Beitrag Leisten gruppieren.

Abbildung 4 zeigt einen Überblick über die taktischen Prozessschritte mit ihren Imperativen und deren Herausforderungen.

[42] eigene Darstellung in Anlehnung an Bukowitz et al. 2002, 269-420.

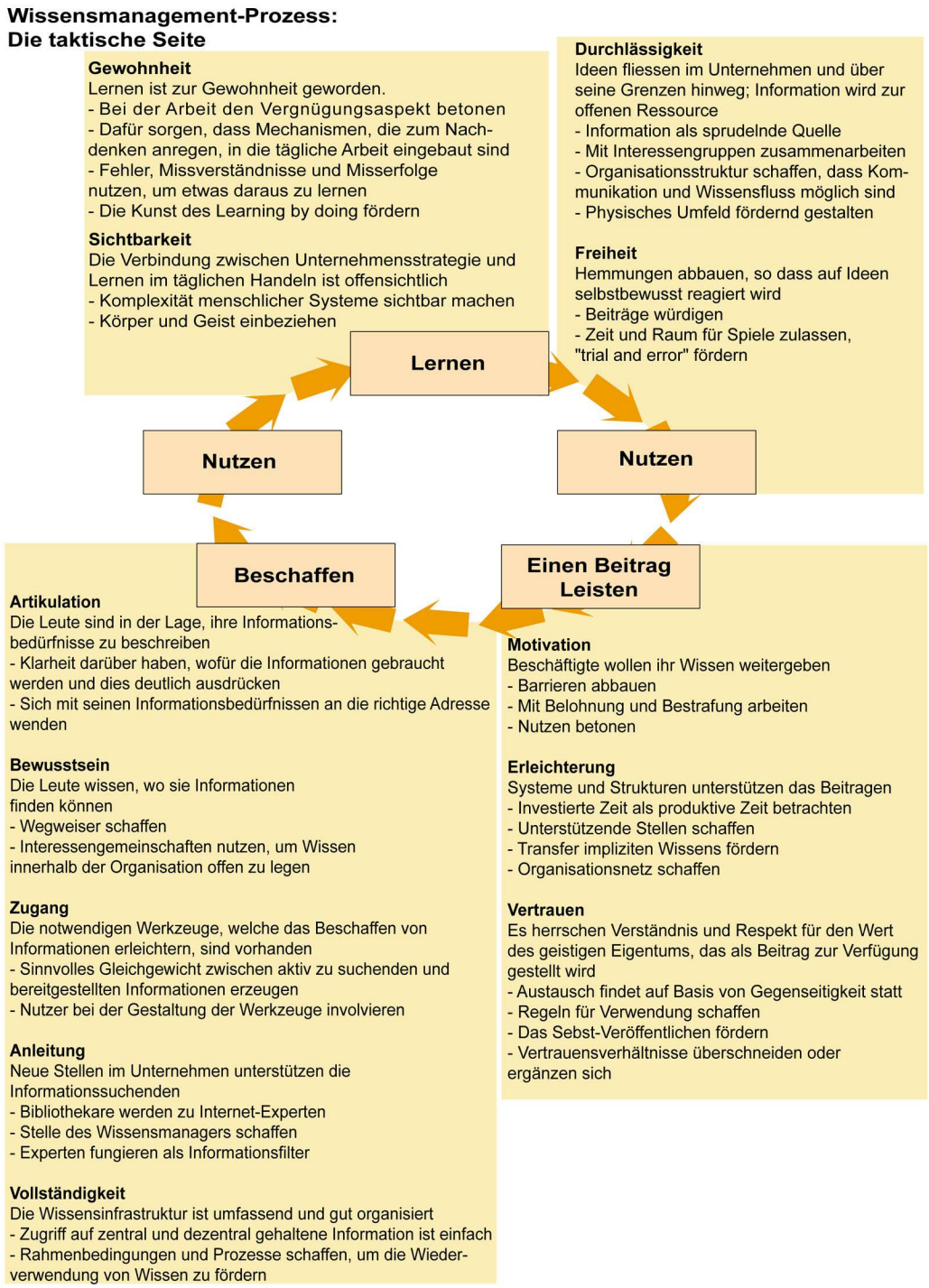


Abbildung 4: Die taktische Seite des Wissensmanagement-Prozesses[43]

Der Kreislauf der taktischen Seite lässt sich wie folgt beschreiben:

"Die Menschen sammeln die Informationen, die sie für ihre tägliche Arbeit brauchen, setzen das Wissen ein, um damit einen Nutzen zu schaffen, lernen aus dem, was sie geschaffen haben, und bringen das neue Wissen zu guter Letzt wieder in das System ein, damit andere es nutzen können, um ihre eigenen Probleme zu lösen."[44] Um diesen Kreislauf zu ermöglichen, sind sowohl Imperative im informationstechnischen Bereich wie auch solche im Bereich des Personalwesens zu beantworten. Das Ziel dabei ist, die Formen der Zusammenarbeit im

[43] eigene Darstellung in Anlehnung an Bukowitz et al. 2002, 47-268.

[44] Bukowitz et al. 2002, 21.

betrieblichen Alltag so zu gestalten, dass sie den Wissensaustausch, die Wissensentwicklung und die Wissensnutzung optimal unterstützen.

2.2 Angewandte Methode

2.2.1 Grundlagen

Um diejenigen Bereiche des Wissensmanagements zu identifizieren, welche für den Softwareprozess besonders wichtig sind, wird das Diagnoseinstrument von Bukowitz et al. 2002, 29-46, verwendet. Es ist einem Fragebogen ähnlich, doch statt um die Beantwortung von Fragen geht es um die Beurteilung von Aussagen. Für jede Aussage entscheidet der Beurteiler, ob für das Unternehmen oder den Unternehmensbereich sehr wichtig, einigermassen wichtig oder nicht wichtig ist, dass sie zutrifft.

Die Aussagen sind in sieben Abschnitte gruppiert. Jeder Abschnitt enthält 20 Aussagen zu einem der Bereiche des Wissensmanagements, wie sie in Abbildung 3 und Abbildung 4 dargestellt wurden.

Die Beurteilungen der Aussagen werden wie folgt gewichtet:

- Anzahl „sehr wichtig“ x 3
- Anzahl „einigermassen wichtig“ x 2
- Anzahl „nicht wichtig“ x 1

Die Summe der gewichteten Nennungen erlaubt zu entscheiden, in welche Bereiche des Wissensmanagements mit Vorrang investiert werden sollte.

Das Diagnoseinstrument ist für die Beurteilung von Unternehmen oder Unternehmensbereichen geschaffen worden. Da in dieser Arbeit aber ein Prozess betrachtet wird, muss es leicht modifiziert werden. Der Diagnosebogen, wie er in dieser Arbeit verwendet wird, ist das Thema der folgenden Abschnitte[45].

2.2.2 Ziel

In Übereinstimmung mit der Fragestellung der Arbeit sollen anhand der Diagnose drei Fragen beantworten werden können:

1. Welche Bereiche des Wissensmanagements sind im Softwareprozess besonders wichtig?
2. Auf welche Bereiche des Wissensmanagements gilt es besonderes Augenmerk zu legen, will man der kollektiven Amnesie in Folge von Entlassungen entgegenwirken?
3. Auf welche Bereiche des Wissensmanagements gilt es besonderes Augenmerk zu legen, wenn die Einarbeitungszeit neu eintretender Mitarbeiter möglichst kurz gehalten werden soll?

2.2.3 Bezugsrahmen

Der Beurteilende ist aufgefordert, sich den für ihn idealen Prozess vor Augen zu halten und die Aussagen bezüglich dieses Ideales zu beurteilen. Es wird aber nicht reichen, den Softwareentwicklungsprozess isoliert zu betrachten. Wie wir oben gesehen haben, lässt sich der Wissensmanagementprozess in einen taktischen und einen strategischen Teil gliedern. In Bezug auf den Softwareprozess beinhaltet der strategische Teil teilweise Massnahmen, welche unternehmensweit umgesetzt werden müssen. Der Beurteilende muss daher den Softwareentwicklungsprozess eingebettet in der gesamten Unternehmenstätigkeit betrachten und damit die Aussagen zusätzlich hinsichtlich ihrer Relevanz für die Softwareentwicklung im Gesamtrahmen des Unternehmens beurteilen.

[45] Ein Exemplar des verwendeten Diagnosebogens kann in Anhang A eingesehen werden.

2.2.4 Modifikation des Diagnoseinstrumentes

Um auf den Softwareprozess anwendbar zu sein, wurde das Diagnoseinstrument verändert[46]. Folgende Fragen mussten beantwortet werden, um zu entscheiden, ob eine Aussage in der Diagnose belassen oder gelöscht wurde:

- Liegt die Realisierung der Aussage im Einflussbereich eines Softwareprozesses?
- Hat die Realisierung der Aussage einen Einfluss auf die Auswahl eines bestimmten Softwareprozesses?

Wurde die Aussage in der Diagnose belassen, so war zu entscheiden, ob sie verändert werden musste. Wo sich eine Aussage auf einen Unternehmensbereich bezog, so wurde sie so verändert, dass sie den Fokus auf den Softwareprozess legt, falls sie im Einflussbereich des Softwareprozesses realisiert werden kann. Hatte die Aussage lediglich einen Einfluss auf die Auswahl des Prozesses, so wurde sie nicht verändert.

2.2.5 Fragestellung

Folgende Fragen waren bei der Beurteilung jeder Aussage zu beantworten:

- Wie wichtig ist es, dass diese Aussage für die Arbeit in einem Softwareprozess zutrifft?
- Wie wichtig ist es, dass diese Aussage für die Arbeit in einem Softwareprozess zutrifft, wenn wir sicherstellen wollen, dass Wissen bei Entlassungen nicht verloren geht?
- Wie wichtig ist es, dass diese Aussage für die Arbeit in einem Softwareprozess zutrifft, wenn wir die Zeit minimieren wollen, die ein neues Teammitglied braucht, bis es sich vollständig eingearbeitet hat?

2.2.6 Vorgehen

Die Befragten waren angehalten, wie folgt vorzugehen:

Der Diagnosebogen sollte möglichst in einem Zug bearbeitet werden. Dies bedeutete einen Aufwand von ungefähr einer Stunde.

Jede Aussage wurde beurteilt, indem die Fragestellungen (vgl. Kapitel 2.2.5) beantwortet wurden. Die Felder im rechten Teil des Fragebogens mussten wie folgt markiert werden:

- „Muss sein“ wurde angekreuzt, wenn die Aussage unbedingt zutreffen muss.
- „Wäre gut“ wurde angekreuzt, wenn es helfen würde, wenn die Aussage zuträfe.
- „Nicht wichtig“ wurde angekreuzt, wenn es keine Rolle spielt, ob die Aussage zutrifft oder nicht.
- „Wäre schädlich“ wird angekreuzt, wenn die Aussage auf keinen Fall zutreffen sollte.

[46] Die Herleitung des Diagnosebogens ist im Anhang B dokumentiert.

2.2.7 Auswertung

Die Auswertung des Diagnosebogens folgt dem in Tabelle 1 gezeigten Schema.

Abschnitt	Anzahl Aussagen	Gewichtung				Maximale Punktzahl
		Muss	Wäre gut	Nicht wichtig	Wäre schädlich	
Beschaffen	15	x 2 * 20/15	x 1 * 20/15	x 0 * 20/15	x –1 * 20/15	40
Nutzen	16	x 2 * 20/16	x 1 * 20/16	x 0 * 20/16	x –1 * 20/16	40
Lernen	19	x 2 * 20/19	x 1 * 20/19	x 0 * 20/19	x –1 * 20/19	40
Einen Beitrag Leisten	15	x 2 * 20/15	x 1 * 20/15	x 0 * 20/15	x –1 * 20/15	40
Beurteilen	17	x 2 * 20/17	x 1 * 20/17	x 0 * 20/17	x –1 * 20/17	40
Aufbauen und Pflegen	12	x 2 * 20/12	x 1 * 20/12	x 0 * 20/12	x –1 * 20/12	40
Aussondern	10	x 2 * 20/10	x 1 * 20/10	x 0 * 20/10	x –1 * 20/10	40

Tabelle 1: Schema der Auswertung des Diagnosebogens

Die Anzahl der Aussagen pro Abschnitt variiert, da, wie in Kapitel 2.2.4 erklärt, gewisse Aussagen aus dem Bogen gestrichen wurden. Um eine egalisierte Gewichtung der Abschnitte zu erhalten, wird das Gewicht jeder Aussage zuerst so erhöht, dass die Anzahl Aussagen multipliziert mit dem Gewicht pro Aussage 20 ergibt. Anschliessend erfolgt eine Multiplikation mit den Faktoren 2 bis –1, welche die Kategorien von Muss bis Wäre Schädlich repräsentieren. Die maximale Punktzahl pro Abschnitt ist dann pro Teilnehmer 40, die minimale –20. Dies ergibt bei 10 Teilnehmern eine maximale Punktzahl pro Abschnitt von 400, eine minimale von –200.

2.3 Die relevanten Wissensmanagementprozessschritte in der Softwareentwicklung

In diesem Abschnitt werden die Ergebnisse der Diagnose dargestellt und diskutiert. Anschliessend wird ein Bild von RUP und XP entworfen, so dass wir dann in der Lage sind, diese beiden Prozessmodelle anhand der Ergebnisse zu beurteilen.

Wie erwähnt wurden die Aussagen zu den Herausforderungen des Wissensmanagements in dreierlei Hinsicht beurteilt:

- bezüglich ihrer Relevanz im Softwareprozess.
- bezüglich ihrer Relevanz, wenn Personen das Projektteam verlassen.
- bezüglich ihrer Relevanz, wenn es um die Einarbeitung neuer Projektmitglieder geht.

In Abbildung 5 bis Abbildung 7 sind die Ergebnisse der Diagnose summarisch dargestellt[47]. Wie leicht zu erkennen ist, wird die Bedeutung jedes einzelnen Schrittes je nach Situation unterschiedlich eingestuft: Während bei Entlassungen und Einstellungen vor allem Wissensmanagement in den Bereichen Einen Beitrag Leisten und Aufbauen und Pflegen notwendig ist, kommt dem Lernen und dem Nutzen in der täglichen Arbeit zusätzlich hohes Gewicht zu. Ebenfalls an Bedeutung gewinnt in dieser Situation der Prozessschritt Aussondern. Während der Einarbeitung neuer Mitarbeiter ist Lernen ebenfalls von grosser Bedeutung.

[47] Die ausführliche Dokumentation befindet sich in Anhang C.

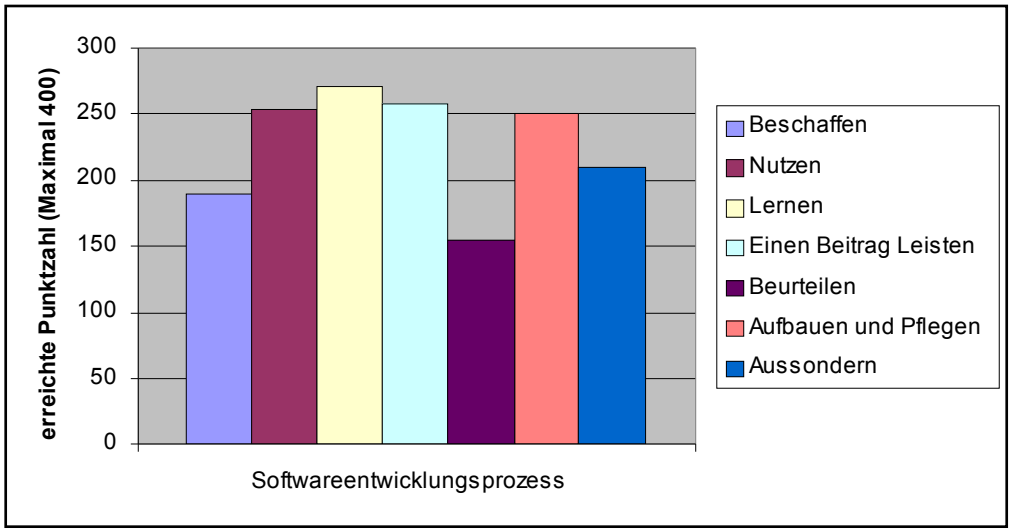


Abbildung 5: Ergebnisse der Diagnose bezüglich der Relevanz der Wissensmanagement-Prozessschritte im Softwareprozess

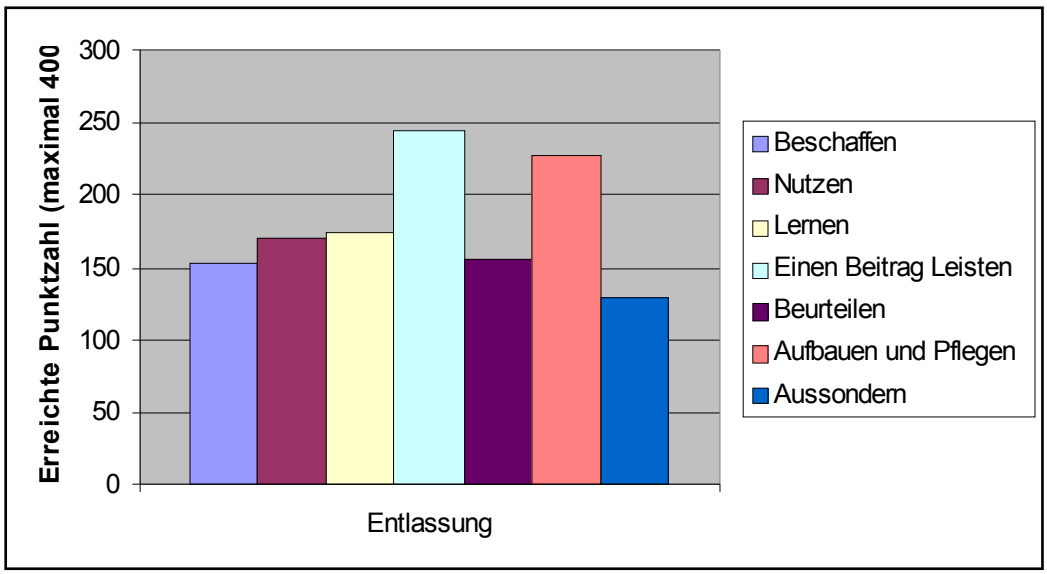


Abbildung 6: Ergebnisse der Diagnose bezüglich der Relevanz der Wissensmanagement-Prozessschritte, wenn Mitarbeiter entlassen werden

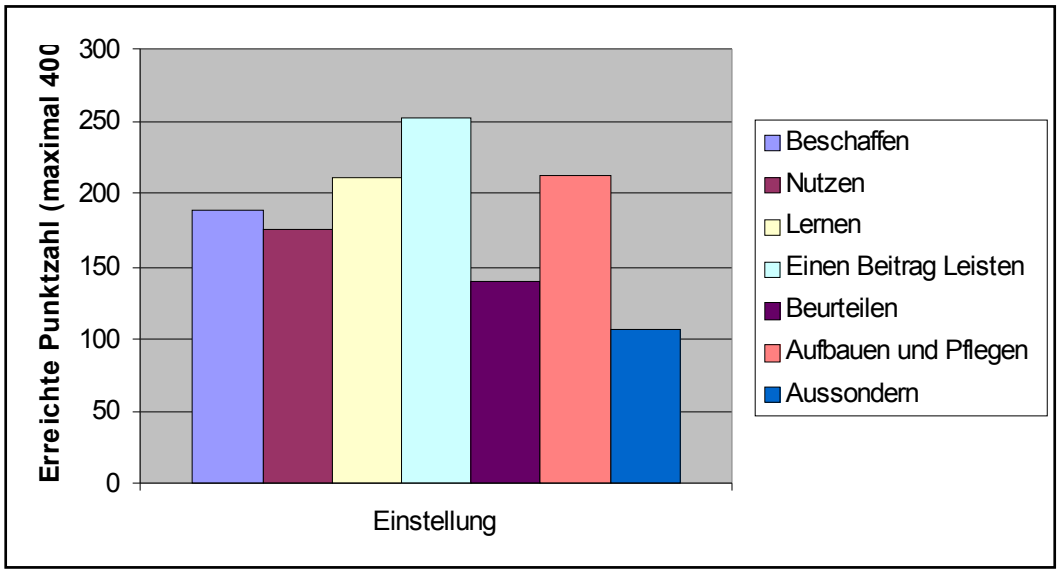


Abbildung 7: Ergebnisse der Diagnose bezüglich der Relevanz der Wissensmanagement-Prozessschritte bei der Einarbeitung neuer Mitarbeiter

Abbildung 8 stellt den Median dar. Die Y-Achse zeigt die erhaltene Punktzahl jedes einzelnen Prozessschrittes als Medianwert der Resultate der 10 befragten Personen. Diese Darstellung hat den Vorteil, dass im Gegensatz zum Durchschnitt extrem abweichende Beurteilungen weniger Gewicht erhalten. Auf der Grundlage der hier ermittelten Werte wird im Kapitel 5 dargestellt werden, wie der RUP und XP die Wissensmanagementbereiche unterstützen.

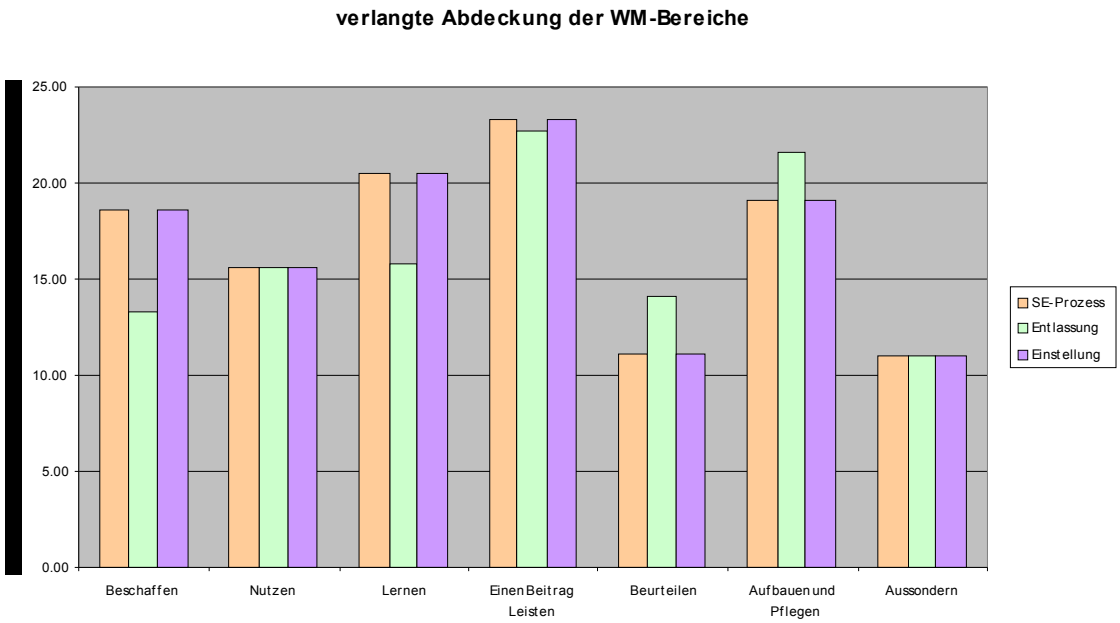


Abbildung 8: Medianwerte der veralngten Abdeckung der WM-Bereiche

Wir erhalten somit die Rangfolge, wie sie Tabelle 2 darstellt. Sie soll in der Folge besprochen werden[48].

Prozessschritt	Rang bzgl. SW-Prozess[1)]	Rang bzgl. Entlassung[2)]	Rang bzgl. Einstellung[3)]	Summe Rang[4)]	Gesamtrang[5)]
Beschaffen	6	5	4	15	5
Nutzen	3	4	5	12	4
Lernen	1	3	3	7	2
Einen Beitrag Leisten	2	1	1	4	1
Beurteilen	7	6	6	19	6
Aufbauen und Pflegen	4	2	2	8	3
Aussondern	5	7	7	19	6

Tabelle 2: Zusammenstellung der Resultate der Diagnose
1): Spalte A, vgl. Abbildung 5.
2): Spalte B, vgl. Abbildung 6.
3): Spalte C, vgl. Abbildung 7.
4): Spalte D = Summe der Werte aus den Spalten A, B und C.
5): Derjenige Prozessschritt erhält den besten Rang, dessen Summe Rang (Spalte D) am kleinsten ist.

Einen Beitrag Leisten befindet sich auf dem ersten Rang in Situationen von Redimensionierungen, auf dem zweiten im Softwareprozess im Allgemeinen. In der täglichen Arbeit wird verlangt, dass jeder sein Wissen anderen zur Verfügung stellt, denn es ist für den Einzelnen unmöglich und auch nicht sinnvoll, sich alle notwendigen Fachkenntnisse und Fähigkeiten anzueignen. Hier ist es von zentraler Bedeutung, dass die Mitarbeiter motiviert sind, ihr Wissen weiterzugeben, dass Systeme und Strukturen bestehen, welche das Beitragen unterstützen und dass die gegenseitigen Beziehungen von Vertrauen, Verständnis und Respekt für den Wert der Beiträge geprägt sind.

Lernen belegt im Softwareprozess Rang 1, in den anderen beiden Situationen Rang 3. Offen zu sein für Neues und ständige Lernbereitschaft zählen zur eigentlichen Kernkompetenz des Softwareentwicklers. Lernen in der täglichen Arbeit muss eine Gewohnheit sein. Günstig wirkt sich hier aus, wenn der Mitarbeiter in der täglichen Arbeit eine Verbindung zwischen seinen Aufgaben und den Zielen des Unternehmens erkennen kann.

Auf Rang drei finden wir Aufbauen und Pflegen. Das Unternehmen sollte über eine solide Wissensbasis verfügen, damit Wissen im Softwareprozess optimal entwickelt und gestreut werden kann. Dies ist besonders wichtig, wenn Personen das Team verlassen und wenn neue Personen dazustossen. Es muss dann die Möglichkeit bestehen, auf eine bewusst gestaltete und gut verwaltete Wissensbasis zurückzugreifen.

Auf Rang vier folgt Nutzen. Die erlernten Fähigkeiten müssen in der täglichen Arbeit umgesetzt werden. Dazu bedarf es einer Kultur, welche Fehler erlaubt, wo die Arbeit nach dem Prinzip von Versuch und Irrtum geschätzt wird und wo frei fliessende Ideen selbstbewusst aufgenommen und ausprobiert werden.

Die Rangfolge wird abgeschlossen von Beschaffen, Beurteilen und Aussondern.

Interessant an den Resultaten der Diagnose ist insbesondere auch die Häufigkeit der Markierungen in der Spalte "wäre schädlich"[49]. Tabelle 3 fasst diese zusammen. Es sind darin nur diejenigen Prozessschritte aufgeführt, deren Total der "wäre schädlich"-Nennungen 6 übersteigt. Für jeden Prozesschritt sind ausserdem nur diejenigen Aussagen erwähnt, deren Total 2 "wäre schädlich"-Nennungen übersteigt. Diese Einschränkung wird eingeführt, um nur diejenigen Aussagen zu besprechen, die von mehreren Befragten als schädlich betrachtet wurden.

[48] vgl. dazu auch Abbildung 3 und Abbildung 4.

[49] vgl. dazu Anhang C.

Prozessschritt und Aussagen	SW-Prozess	Entlassung	Einstellung	Total
Beschaffen				39
Dabei gibt es solche Rollen, die sich vorwiegend um die administrative Arbeit kümmern und solche, die sich auf die Inhalte konzentrieren.	2	2	1	5
Die Leute fordern nur dann Informationen, wenn sie diese wirklich brauchen.	2	2	1	5
Experten helfen bei der Identifizierung von wichtigen Informationen für andere Mitarbeiter.	2	0	1	3
Die Leute können Informationen in vielen verschiedenen Datenbanken und Anwenderprogrammen suchen.	4	4	4	12
Wir unterscheiden zwischen Informationen, die zentral verwaltet werden sollten, und solchen, die jeder individuell aufzeichnen und an andere weitergeben kann.	4	5	5	14
Beurteilen				12
Im Unternehmen wird unser Wissensmanagement-Prozess und dessen Ergebnisse gemessen.	1	1	1	3
Wir können die Aktivitäten im Rahmen des Wissensmanagements zu messbaren Resultaten in Beziehung setzen.	1	1	1	3
Die Leute wissen, welche Massstäbe verwendet werden, um den Wissensmanagement-Prozess und dessen Ergebnisse zu beurteilen.	1	1	1	3
Aussondern				8
Wir holen Fähigkeiten und Fachkenntnisse von aussen herein, wenn sie nicht zu unseren Kernkompetenzen gehören.	2	1	1	4

Tabelle 3: Resultate der Diagnose: Aussagen mit der Markierung "wäre schädlich"

Insbesondere Instrumente des Prozessschrittes Beschaffen sind umstritten. Abgelehnt wird der Gedanke, dass verschiedene Rollen für unterschiedliche Aufgabenbereiche des Wissensmanagements entstehen, aber auch, dass Experten bei der Informationssuche helfen. Hinderlich wird auch empfunden, wenn die Informationen in vielen unterschiedlichen Datenbanken gesucht werden müssen und wenn neben zentralen auch dezentrale Informationsspeicher existieren.

Ein weiterer Prozessschritt, der gefährlich erscheint, ist Beurteilen. Nach einem persönlichen Gespräch mit dem Beurteilenden hat sich ergeben, dass das Messen deshalb als hinderlich betrachtet wird, weil es das Gefühl vermittelt, überwacht zu werden. Die Freiheit, sich für Dinge zu interessieren, die mit der unmittelbaren Arbeit nichts zu tun haben, wäre eingeschränkt.

Der dritte kritische Bereich ist Aussondern. Die Beurteilenden sind teilweise der Meinung, dass kein Wissen von ausserhalb der Firma beansprucht werden sollte. Sie geben zu bedenken, dass die Fachkenntnisse für die Erstellung der Softwarelösungen im Team selbst aufgebaut werden müssen.

Anhand dieser Resultate sollen die beiden Softwareprozesse RUP und XP beurteilt werden. Um für diese Beurteilung die Grundlagen zu legen, werden in den beiden folgenden Kapiteln die beiden Prozesse skizziert.

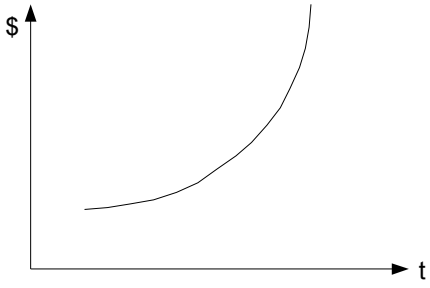
3 Extreme Programming

3.1 Überblick

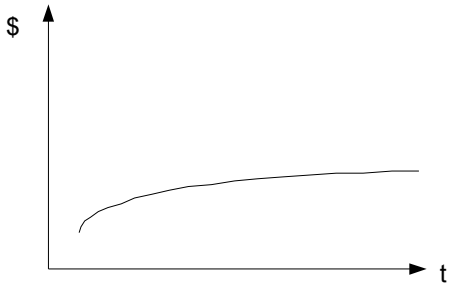
Extreme Programming, kurz XP, ist ein Vorgehensmodell, das für kleine bis mittlere Softwareprojekte geeignet ist, die in einem vagen Umfeld abgewickelt werden und bei denen Anforderungen häufig ändern.[50] Die Voraussetzung dafür, dass dieses explorative, flexible Modell Erfolg verspricht, ist die Ungültigkeit der für lange als gegeben betrachteten exponentiellen Kostenkurve für Änderungen (vgl. Abbildung 9A) [51].

[50] Dieser Abschnitt stützt sich auf Beck 2000.

[51] vgl. Beck 2000, 21-25. RUP hält an der Gültigkeit der traditionellen Kostenkurve fest (vgl. Kruchten 2000, 12f und 26).



A: Exponentielle Kosten von Änderungen mit dem Vergehen von Zeit macht es sinnvoll, heute Geld auszugeben, um die Kosten von später zu sparen.



B: Asymptotischer, flacher Kostenverlauf bei Änderungen spricht dafür, heute nur soviel zu bezahlen, wie absolut notwendig ist und für Änderungen später aufzukommen.

Abbildung 9: Der Kostenverlauf für Änderungen[52]

Der Verlauf der Kostenkurve wie sie in Abbildung 9A dargestellt ist, ist nicht mehr gültig, denn die Bemühungen der letzten Jahre, bessere Programmiersprachen und -praktiken, Datenbanktechnologien und Werkzeuge zu entwickeln mit dem Ziel, die Kosten von Änderungen zu senken, haben Früchte getragen: Mit der objektorientierten, werkzeugunterstützten Entwicklung gilt heute eine asymptotische Kostenkurve, wie sie in Abbildung 9B dargestellt ist. Diese neue Kostenkurve legt nahe, die kritischen Entscheidungen so spät als möglich im Prozess zu treffen, um die Kosten der Entscheidung so spät als möglich tragen zu müssen. Ebenfalls wird man keine Programmteile mehr implementieren, die für die zur Zeit gewünschte Funktionalität nicht wichtig sind. Designveränderungen werden immer dann eingeführt, wenn dadurch existierende Komponenten vereinfacht oder neu hinzukommende Anforderungen einfacher implementiert werden können. Auf diesem Hintergrund wird eine Form der Zusammenarbeit gezeichnet, die sich auf provokative Art von den bisherigen Projektabläufen unterscheidet.

Abbildung 10 zeigt den Ablauf eines XP-Projektes. Am Anfang des Projektes und jedes Releases werden `User Stories`[53] verfasst und es wird ein `Architectural Spike` erstellt. Dies ist eine Grobsicht der Architektur, für die ein kleiner Prototyp erstellt und eine `Metaphor`[54] formuliert wird. Diese beiden Elemente bilden die Eingangsgrössen für das `Release Planning`. Falls während des Planens eine `Story` nur sehr unsicher geschätzt werden kann, zum Beispiel weil man etwas Ähnliches noch nie gemacht hat, so wird ein `Spike` erstellt. Dies ist ein experimenteller Lösungsversuch, mit dem die Komplexität des Problems abgeschätzt werden kann[55].

Der in der Folge zu erstellende Releaseplan legt fest, was in der `Iteration`, dem Zeitraum bis zum nächsten Release, implementiert werden soll. Jede `Iteration` wird abgeschlossen mit einem `Acceptance Test`, der von Benutzern oder automatisiert anhand von `Test Scenarios` durchgeführt wird. `Test Scenarios` werden aus den `User Stories` abgeleitet. Falls der Benutzer mit den Testergebnissen zufrieden ist, wird der Release freigegeben, andernfalls wird die Verbesserung der `Story` in die nächste `Iteration` aufgenommen.

[52] vgl. Beck 2000, 21-25.

[53] Eine User Story ist vergleichbar mit einem Use Case.

[54] vgl. Kapitel 3.4.

[55] vgl. Jeffries et al. 2001, 41f.

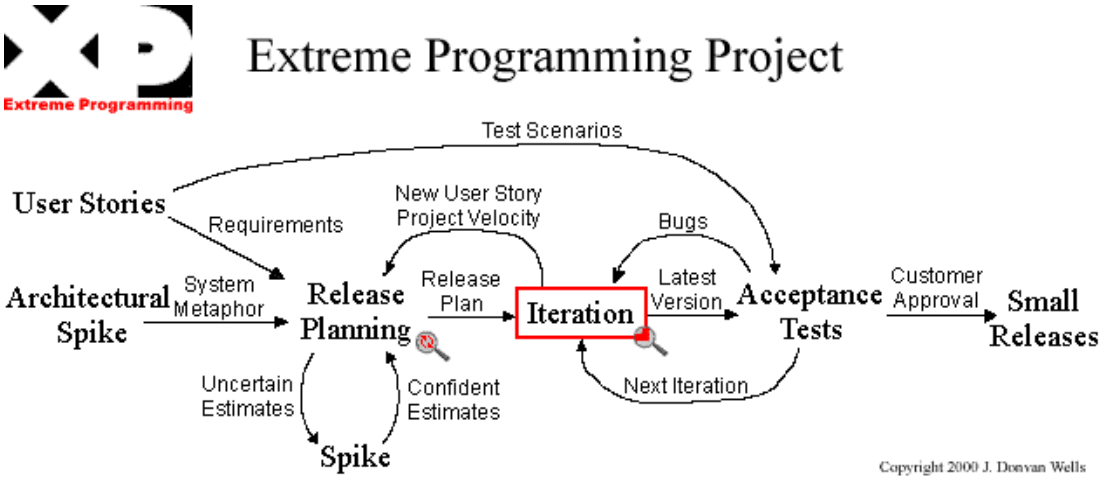


Abbildung 10: Ablauf eines XP-Projektes[56]

In den folgenden Kapiteln sollen die Eigenheiten dieses Prozesses dargestellt werden, um anschliessend eine Beurteilung bezüglich der in Kapitel 2.3 ermittelten Wissensmanagement-Aktivitäten vorzunehmen. In dieser Absicht werden zuerst die Grundgedanken erklärt, die hinter XP stehen. In der Folge wird ein Verständnis für die Grundwerte der Zusammenarbeit und die unter diesem Gesichtspunkt sinnvollen Praktiken und Rollenverteilungen erarbeitet.

3.2 Die vier Variablen

In der Softwareentwicklung gibt es vier Variablen: `Costs` (Kosten), `Time` (Zeit), `Quality` (Qualität) und `Scope` (Umfang). Diese vier Grössen beeinflussen das Projekt und können zu Zielkonflikten führen. Alle vier Variablen sollen durch den Prozess sichtbar gemacht werden. Dies erlaubt es, den Prozess durch Veränderung der Variablen zu steuern[57].

Die vier Variablen beeinflussen sich gegenseitig wie folgt[58]:

- `Costs` – wird mehr Geld in ein Projekt investiert, so kann dies das Tempo der Entwicklung beschleunigen, kann es aber auch bremsen.
- `Time` – ein Mehr an Zeit kann bewirken, dass mehr Funktionalität in besserer Qualität realisiert wird. Allerdings kann die Dauer zwischen den Integrationen zu lange sein, was der Qualität wiederum schadet. Zu wenig Zeit wirkt sich negativ auf Qualität, Umfang und Kosten aus.
- `Quality` – Qualität ist die heikelste Variable. Ihre Beeinflussung sollte unterlassen werden, denn die Gewinne sind höchstens kurzfristiger Natur. Mittelfristig gesehen wirkt sich eine Verminderung der Qualität negativ auf die Kosten, die Termine und den realisierbaren Umfang aus.
- `Scope` – Weniger Umfang erlaubt höhere Qualität und/oder frühere Termine. Zu viel Umfang setzt die Projektmitglieder unter Druck, die Qualität leidet und Termine können unter Umständen nicht gehalten werden, was auch die Kosten in die Höhe treibt.

Leider ist das Muster der gegenseitigen Beeinflussung der vier Grössen kein einfaches. Beck empfiehlt daher, nur an der Variablen `Scope` Anpassungen vorzunehmen, da diese am leichtesten zu steuern ist und die Auswirkungen ihrer Veränderung am ehesten durchschaubar sind[59]. Termine sind eine Grösse, die in XP grundsätzlich nie verändert wird. Bezüglich der Qualität sollten aus den oben genannten Gründen keine Kompromisse eingegangen werden. Kosten können verändert werden, zum Beispiel indem das Team vergrössert oder verkleinert

[56] von http://www.extremeprogramming.org/map/project.html (am 7. Juni 2003).

[57] Dies ist Aufgabe des Planungsprozesses, der in Kapitel 3.4, Seite 21, besprochen wird.

[58] vgl. Beck 2000, 16.

[59] vgl. Beck 2000, 18f.

wird. Eine Vergrösserung führt aber oft zu einer Verlangsamung des Projektverlaufes, während eine Reduktion zu Know-how-Verlust führt.

Diese Überlegungen enden in der Empfehlung, dass der Kunde über den Umfang entscheidet, während die Entwickler die benötigte Zeit bestimmen. Umfang und benötigte Zeit werden in der Planungssitzung auf die Iterationen abgestimmt. Termine und Qualität sind unveränderliche Variablen.

3.3 Die Grundwerte

Alle Entwickler müssen vier Grundwerte akzeptieren, damit die Gruppe zum Ziel kommt[60]. Hier wird die Bedeutung des Wortes `extreme` deutlich: Verinnerlicht nicht jeder Entwickler diese Grundwerte, wird das mit XP durchgeführte Projekt nicht erfolgreich sein.

Diese Grundwerte sind die folgenden:

- `Communication` – Probleme im Projekt sind häufig darauf zurückzuführen, dass jemand etwas Wichtiges nicht mit jemand anderem besprochen hat. Dies ist beispielsweise der Fall, wenn ein Entwickler einen anderen Entwickler über eine wichtige Designveränderung nicht informiert oder wenn der Entwickler dem Kunden bezüglich einer Anforderung nicht die richtige Frage stellt. Für mangelnde Kommunikation gibt es tausend Gründe. Beispielsweise wird ein Entwickler, der dem Manager schlechte Nachrichten überbringt und dafür bestraft wird, mit dem Manager nicht mehr offen sprechen und ein Kunde, dessen wichtiger Hinweis ignoriert wurde, sich künftig die Mühe nicht mehr geben, dem Entwickler viel zu erklären. XP setzt sich daher zum Ziel, durch die eingesetzten Praktiken die Kommunikation zwischen den Beteiligten zu fördern und zu öffnen.
- `Simplicity` – Es gilt die Regel, immer die einfachste Variante zu wählen. Etwas Bestehendes, das vereinfacht werden kann, wird vereinfacht.
- `Feedback` – Feedback bezieht sich sowohl auf die Rückmeldungen des Systems durch Tests als auch auf diejenigen der Endbenutzer. Möglichst rasches Feedback ist in XP zentral, denn dadurch wird der Lernprozess gefördert und der Projektfortschritt beschleunigt.
- `Courage` – Mut ist die Voraussetzung für ein Leben mit XP. Ohne Mut wird weder ein Systemdesign mit dem Ziel der Vereinfachung umgestellt, noch werden kritische Worte ausgesprochen oder eine neue Funktionalität schnell integriert, damit der Anwender Rückmeldung darüber geben kann.

3.4 Die Praktiken

XP bringt die Grundwerte in den folgenden Praktiken zur Anwendung[61]:

- `The Planning Game` – Für das Erstellen der Planung in XP gelten folgende Regeln:

- Die Teammitglieder melden sich freiwillig, eine Aufgabe zu übernehmen.
- Wer für die Realisierung verantwortlich ist, schätzt den Aufwand.
- Abhängigkeiten zwischen den Aufgaben werden nicht berücksichtigt, da XP davon ausgeht, dass es keine solchen gibt.
- Die Reihenfolge der Realisierung folgt der Priorität der `User Stories`.
- Die detaillierte Planung bezieht sich auf eine Iteration. Für die darauf folgende Zeit wird lediglich eine Grobplanung geführt.

[60] nachzulesen in Beck 2000, 29-35.

[61] Ausführungen zu den Praktiken finden sich unter anderem in Beck 2000, 54-70.

Wie aus diesen Regeln ersichtlich, muss die Planung im Team stattfinden. Als Methode wird das sog. Planning Game angewendet, zu dem alle Teammitglieder inklusive sämtliche Kunden, die über die zu realisierenden Funktionalitäten entscheiden können, zusammenkommen. Gespielt wird mit Story Cards. Sie enthalten eine Kurzbeschreibung der User Stories. Das Ziel des Spieles ist es, die verfügbare Zeit der Entwickler so zu verplanen, dass der maximal mögliche Kundennutzen entsteht. Während der Kunde für die Priorisierung der Anforderungen zuständig ist, schätzen die Entwickler die Aufwände für deren Realisierung. Die verfügbare Zeit der Iteration wird dann mit Anforderungen aufgefüllt.

Während der Iteration wird die Planung rollend geführt. Ein Festhalten an der ersten Planung kann ein Projekt in den Abgrund treiben, denn dies hätte zur Folge, dass man weder weiss, was tatsächlich an Arbeit noch zu erledigen ist, noch was die bereits implementierte Funktionalität beinhaltet. Mit Planen verfolgt man aber gerade das Ziel, jederzeit das wichtigste noch zu Erledigende zu tun, effektiv mit anderen zu koordinieren und schnell auf unerwartete Ereignisse reagieren zu können[62].

Das Planen gewinnt für XP umso mehr an Bedeutung, als eben nicht nach einem Phasenmodell vorgegangen wird. Das flexible Arbeiten in der explorativen Entwicklung braucht die rollende Planung, um den Projektfortschritt transparent zu halten.

- Small Releases – Die einzelnen Releases sollen in so kurzen Abständen wie möglich ausgeliefert werden. Der Benutzer soll so möglichst früh mit der Realität des entwickelten Systems konfrontiert werden und Feedback geben können. Die Anforderungen werden so in vielen kurzen Releasezyklen in der Reihenfolge ihrer Priorität implementiert. Dies hat den Vorteil, dass der Kunde nicht zu Beginn des Projektes entscheiden muss, welche Anforderungen das System erfüllen muss. Er kann von Release zu Release Anforderungen hinzufügen und wegnehmen.
- Metaphor – Jede Applikation sollte mit Hilfe der Metaphor eine konzeptuelle Integrität erhalten, die sich in der begrifflichen Ausdrucksweise auswirkt. Das Ziel der Methaphor ist es, ein gemeinsames Verständnis darüber zu erhalten, wie und was das System ist. Als Beispiel mag das Folgende dienen: "C3 payroll, ..., is like a manufacturing line, building paychecks by taking hour parts and converting them to dollar parts, assembling all the various parts until you get a paycheck."[63]
- Simple Design – Das richtige Design einer Software zu einem bestimmten Zeitpunkt ist dasjenige, welches

- sicherstellt, dass alle Tests erfolgreich durchlaufen.
- keine duplizierte Logik enthält.
- sich selbst erklärt.
- die geringst mögliche Anzahl Klassen und Methoden hat.

- Testing – Das Testing ist das Herzstück von XP. Ein Programmteil, für das kein automatisierter Test besteht, wird in XP als nicht existierend betrachtet. Der Unittest wird vom Programmierer vor dem Schreiben der Funktionalität verfasst. So soll das Vertrauen, das der Entwickler in sein Stück Code hat, zu einem Teil des Codes selbst werden. Ausserdem soll dadurch verhindert werden, dass kompliziertes Design entsteht, ist es doch schwierig, für komplizierte Sachverhalten einen Test zu schreiben.
  Die Kunden erstellen automatisierte Funktionstests für jede User Story, so dass Abnahmetests reproduzierbar bleiben. Die Tests sollen so neben dem Code selbst zur Spezifikation der Software werden, indem sie nicht nur das gesamte Verhalten des Systems überprüfen, sondern es auch beschreiben.

[62] vgl. Beck et al. 2000, 1.

[63] Jeffries et al. 2001, 80.

Das Testen ist in XP eine in den Entwicklungsprozess integrierte Tätigkeit und nicht eine separate Aufgabe oder gar eine isolierte Phase. Dies setzt voraus, dass die Tests genauso wie der Code weiterentwickelt werden. Sie sind stets ein Spiegel des aktuellen Projektfortschrittes.

- `Refactoring` – Refactoring bedeutet, bestehenden Code immer wieder zu verändern, um ihn so einfach wie möglich zu machen[64]. Nachdem die Entwickler eine neue Funktionalität hinzugefügt haben, schauen sie das Design noch einmal an und vereinfachen es, auch wenn dies kurzfristig einen gewissen Mehraufwand bedeutet.
- `Pair Programming` – Jede Entwicklungsarbeit wird von zwei Personen mit einem Bildschirm, einer Tastatur und einer Maus ausgeführt. Einer der beiden Entwickler ist der `Driver`. Er bedient die Tastatur und überlegt sich, wie die Funktionalität am besten implementiert werden kann. Der andere, der `Partner`, bewegt sich in seinen Gedanken auf der strategischen Seite und fragt sich zum Beispiel, ob der gewählte Ansatz sinnvoll ist, ob es neue Testfälle gibt oder ob die Möglichkeit besteht, den ganzen Code zu vereinfachen.
  Von dieser Vorgehensweise verspricht man sich weniger Fehler, einen regen Know-how-Austausch und einfaches Design. Die Verfechter von XP sind sogar der Überzeugung, dass mit `Pair Programming` schneller entwickelt wird als wenn jeder für sich alleine codiert.
- `Collective Code Ownership` – Der Code gehört grundsätzlich allen Teammitgliedern und jeder im Team ist für den gesamten Code verantwortlich. Jeder hat die Aufgabe, bestehenden Code zu verändern, sobald er Verbesserungsmöglichkeiten erkennt.
- `Continuous Integration` – Sobald eine neue `User Story` oder ein Teil davon, der dem Benutzer Mehrwert schafft, fertig ist, wird sie ins System integriert. Somit erhalten die Anwender ein Bild der neuen Funktionalität und können gegebenenfalls erwachende Änderungsbedürfnisse umgehend anbringen.
- `40-Hour Week` – In XP wird mit der Überzeugung gearbeitet, dass nur ausgeruhte Teammitglieder qualitativ hochstehende geistige Arbeit verrichten können[65]. Überstunden können notwendig werden, dürfen aber nicht über länger als eine Woche geleistet werden. Chronisches Anfallen von Überstunden ist ein klares Zeichen für eine Projektkrise.
- `Onsite Customer` – Ein späterer Benutzer sollte seinen Arbeitsplatz zum Entwicklungsteam verlegen. Dies fördert den Austausch zwischen Entwicklern und Kunden. Fragen können unmittelbar beantwortet werden. Trotzdem kann der Kunde seinen regulären Tätigkeiten nachgehen.
- `Coding Standards` – Ein minimaler Satz an Regeln bezüglich der Codierung muss von allen Teammitgliedern befolgt werden. Dies ist eine Voraussetzung für das Funktionieren der Praktiken `Pair Programming`, `Collective Code Ownership` und `Refactoring`.

Abbildung 11 gibt einen Überblick über das Zusammenwirken verschiedener Praktiken.

[64] Über `Refactoring`-Praktiken existieren bereits zahlreiche Publikationen. Die bekannteste ist Fowler 1999.

[65] Ein vehementer Verteidiger dieser Auffassung ist auch De Marco 2002, 45-132 und De Marco 1998, 18.

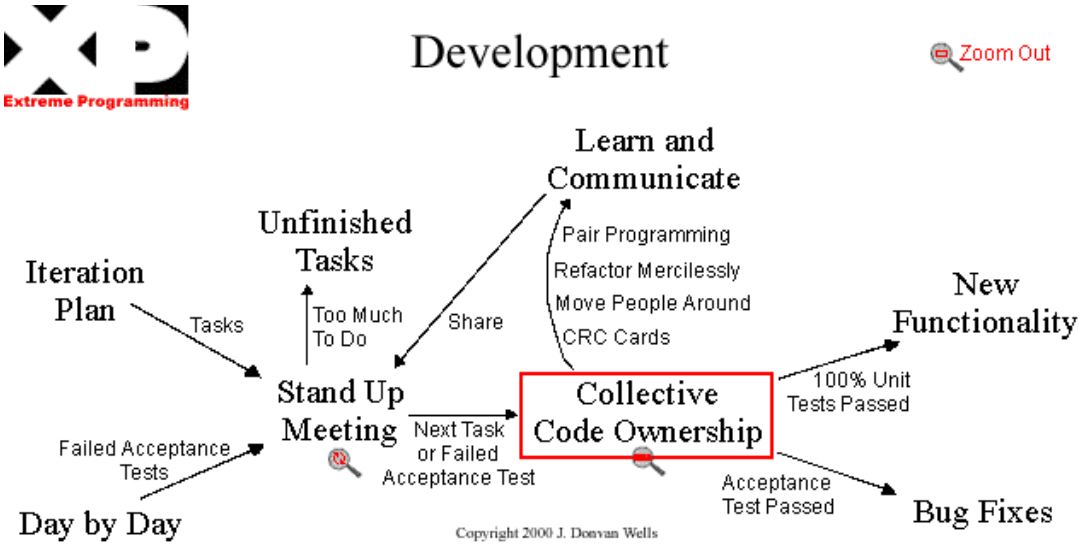


Abbildung 11: Tätigkeiten während der Entwicklung mit XP[66]

Im Zentrum der Entwicklung steht die Praxis des `Collective Code Ownership`, welche impliziert, dass sich jedes Teammitglied für alle Bestandteile des Codes verantwortlich fühlt.

Täglich findet das `Stand Up Meeting`[67] statt: Die Entwickler stehen für das kurze Meeting in einen Kreis und erzählen, woran sie gerade arbeiten und wo Probleme aufgetaucht sind. Im `Stand Up Meeting` werden Probleme lediglich zur Sprache gebracht, sie werden nicht dort gelöst. Die Personen, die für die Lösung Ideen haben, arbeiten an dem Tag als Pair zusammen. So wird einerseits sichergestellt, dass innerhalb des Entwicklerteams Erfahrungen genutzt werden können, andererseits weiss nach dem Meeting jeder im Team um den Status des Projektes Bescheid.

Wie wir sehen, ist `Learn and Communicate` ein wichtiger Bestandteil des Entwicklungsprozesses. Der Lernprozess wird unterstützt durch `Pair Programming`, wo Teammitglieder immer wieder in neuen Paaren zusammenarbeiten, und `Refactoring`.

Die Entwicklung mit XP setzt geeignete Räumlichkeiten voraus. `Pair Programming` verlangt nach grossen Arbeitsplätzen, die flexibel getauscht werden können. Ausserdem ist der Einsatz von Werkzeugen, welche den raschen Zyklus von Erstellen, Testen und Integrieren von Programmcode ermöglichen, vorteilhaft.

3.5 Die Rollenverteilung

Ein Projektteam lässt sich mit einer Mannschaft im Sport vergleichen[68]. Jeder Spieler einer Fussballmannschaft beispielsweise übernimmt mit seiner Position einen Verantwortungsbereich. Mit dem Verantwortungsbereich übernimmt jeder auch eine Rolle. Die Rollen werden akzeptiert und wahrgenommen, weil sie funktional richtig sind. Doch sie sind nicht starr – vielmehr werden sie durch die Mannschaft und den Coach aktiv gestaltet. Analog gibt es in der Softwareentwicklung Verantwortungsbereiche und damit Rollen. XP kennt die folgenden:

- `Programmer` – Die zentrale Rolle im Entwicklungsprojekt ist der `Programmer`. Jemand in dieser Rolle muss neben seinem Fachwissen einige persönliche Merkmale mitbringen: Er muss kommunikativ, teamorientiert, kooperativ und mutig sein. Die Rechte und Pflichten des `Programmers` sind in der `Bill of Rights`

[66] von http://www.extremeprogramming.org/map/development.html am 23.6.2003.

[67] Genaueres über das `Stand Up Meeting` ist in Beck et al. 2000, 105-106 zu finden.

[68] Dieser Abschnitt folgt Beck 2000, 139-148.

zusammengefasst[69]: Er hat das Recht, zu wissen was vom System verlangt ist, Qualitätsarbeit zu leisten und Hilfe von Kollegen und Fachvertretern anzufordern. Er darf die Aufwände selbst schätzen und die Schätzungen korrigieren sowie Verantwortung freiwillig übernehmen.

- `Customer` – Während der `Programmer` weiss, wie man programmiert, weiss der `Customer`, was programmiert werden soll. Der `Customer` muss über Prioritäten von Anforderungen entscheiden können. Ausserdem muss er in der Lage sein, `User Stories` zu verfassen und funktionale Tests zu schreiben. Der `Customer` muss `Real` sein, dass heisst, er darf nicht zu einer Zwischeninstanz gehören, die zwischen Entwicklern und Anwendern vermittelt. Auch seine Rechte sind in der `Bill of Rights` beschrieben[70]: Er hat das Recht, einen Projektplan einzusehen und aus der Arbeit der `Programmer` den maximalen Nutzen zu erhalten. Er hat das Recht, seine Meinung zu Anforderungen und Prioritäten zu ändern.
- `Tester` – Der `Tester` trägt die Verantwortung dafür, dass der `Customer` und die `Programmer` sämtliche Tests regelmässig durchführen. Zudem unterstützt er den `Customer` bei der Durchführung der funktionalen Tests.
- `Tracker` – Als Terminmanager fragt der `Tracker` die Teammitglieder regelmässig nach ihren Fortschritten, protokolliert die Testergebnisse und kann stets über den Projektstatus Auskunft geben.
- `Coach` – Der Prozess wird vom `Coach` überwacht, der ein tieferes Verständnis von XP mitbringt und mit anderen XP-Teams im Austausch steht. Er muss intervenieren, wenn er beispielweise zur Überzeugung gelangt, dass jemand im Team stört, oder dass eine falsche Richtung eingeschlagen wurde. Ansonsten sollte er den ruhigen Pol verkörpern.
- `Consultant` – Durch `Pair Programming` und den Mut der Programmierer, Verantwortung zu übernehmen, werden in einem XP-Projekt keine schwarzen Löcher entstehen, wo sich niemand auskennt. Das Team wird so extrem flexibel. Trotzdem kann es vorkommen, dass fundiertes technisches Verständnis ausserhalb des Teams gesucht werden muss. In diesem Falle wird ein `Consultant` eingestellt, der sich an die Arbeitsweise des Teams anpassen muss. Neben der Problemlösung wird von ihm verlangt, dass er die Teammitglieder so schult, dass sie ein solches Problem in Zukunft selbst lösen können. Mindestens ein anderes Teammitglied verrichtet die Arbeit mit ihm, so dass die Vorteile des `Pair Programmings` ausgeschöpft werden.
- `Big Boss` – Dem Team Mut und Vertrauen zusprechen und wichtige Entscheidungen treffen – das ist die Aufgabe des `Big Boss`. Er muss offen kommunizieren und mit ehrlichen Aussagen zurechtkommen können. Als Mentor steht er mit vollstem Vertrauen hinter dem Team und stützt dessen Entscheidungen und Vorgehensweisen.

Nachdem wir nun einen Überblick über die Grundsätze von XP erhalten haben, wird im den nächsten Abschnitt der RUP in seinen Grundzügen dargestellt. Dies wird den Vergleich der beiden Prozessmodelle ermöglichen.

[69] vgl. Beck 2000, 147f, Jeffries et al. 2001, 7.

[70] vgl. Beck 2000, 147f, Jeffries et al. 2001, 7.

4 Der Rational Unified Process

4.1 Überblick

Der Rational Unified Process gehört zu den iterativen Vorgehensmodellen. Er wurde 1995/6 von Ivar Jacobson, Grady Booch und James Rumbaugh bei der Firma Rational Software entwickelt[71].

Der RUP sieht die vier Phasen `Inception` (Konzeptualisierungsphase), `Elaboration` (Entwurfsphase), `Construction` (Konstruktionsphase) und `Transition` (Auslieferung des Systems an den Kunden) vor.

Im Unterschied zum Wasserfallmodell, dessen augenfälligste Eigenschaft ja ebenfalls die Unterteilung des Projektes in Phasen ist[72], werden im RUP innerhalb der Phasen Iterationen vorgesehen. Ausserdem erstrecken sich einzelne Tätigkeitsfolgen – im RUP `Workflows` genannt – über die gesamte Projektdauer, während im Wasserfallmodell die Tätigkeiten einer Phase mit dem Ende der Phase abgeschlossen sind.

zeigt einen Überblick über das Vorgehen. Die beiden Achsen repräsentieren die beiden dynamischen Dimensionen der Prozessstruktur:

- Die Horizontale ist die Zeitachse, auf der die Phasen sequentiell ablaufen. Jede Phase wird durch einen Meilenstein abgeschlossen.
- Auf der Vertikalen sind die verschiedenen `Workflows` dargestellt, welche die verschiedenen Aktivitäten logisch gruppieren. Je nach Dicke des Streifens erfordert der jeweilige `Workflow` mehr oder weniger Aktivität.

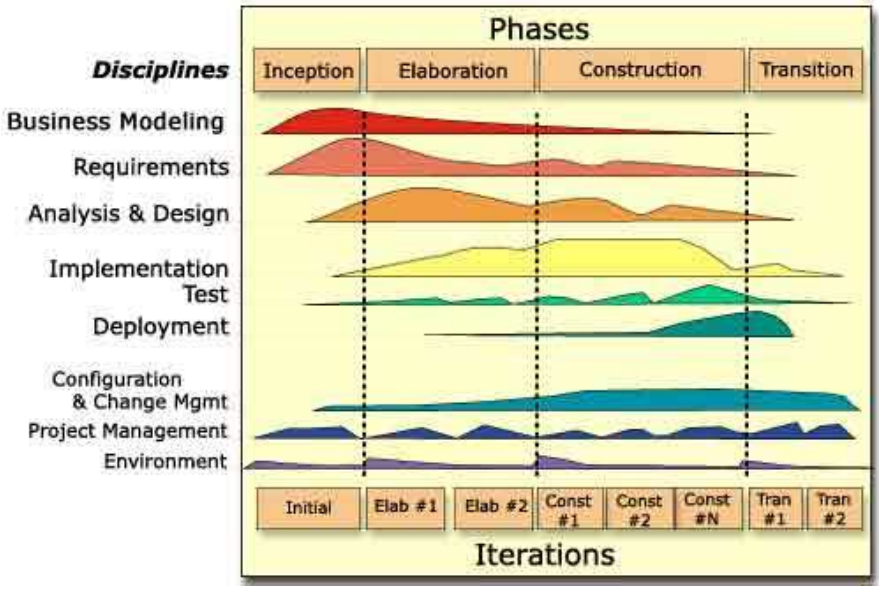


Abbildung 12: Die beiden Dimensionen des RUP[73]

In der Folge wird dargestellt, welches die deklarierten Ziele des RUP sind und unter Verwendung welcher Prinzipien er diese erreichen möchte. Anschliessend werden die Elemente beschrieben, die sicherstellen sollen, dass der Prozess seinen geordneten Gang nimmt: Jede Aktivität jedes `Workflows` muss zur richtigen Zeit von der richtigen Person ausgeführt werden und den richtigen

[71] Das folgende Kapitel folgt Kruchten 2000, Versteegen 2000 und dem eCoach von RUP (in der Folge wird darauf in der Form von "RUP 2000 eCoach" verwiesen), von wo auch alle Abbildungen übernommen wurden. Eine Version dieses eCoaches kann heruntergeladen werden von www.rational.com/products/rup (am 23. 6. 2003)

[72] vgl. Sommerville, 2001, 45-46.

[73] von http://www.rational.com/global/includes/bigimage.jsp?bigimage_path=/images/products/rup/rup_poster_lg.jpg&refering_page=/products/rup/prodinfo.jsp (am 23.6.2003)

Output entstehen lassen. Die folgenden Schilderungen sollen das Fundament für die Untersuchung legen, wie den Herausforderungen des Wissensmanagements mit RUP begegnet werden kann.

4.2 Ziele

Mit dem RUP besteht ein disziplinierter Zugang zum modernen Projektmanagement. Das Ziel des RUP ist es sicherzustellen, dass Software in hoher Qualität zu planbaren Kosten und zeitlichen Aufwänden hergestellt wird, und dass diese Software die Bedürfnisse der Endbenutzer abdeckt. Die Autoren des RUP beanspruchen für sich, die Ursachen der Softwarekrise erkannt zu haben[74]. Diese macht sich seit den 90er Jahren bemerkbar und äusserte sich im Jahr 1995 beispielsweise darin, dass 31% aller Softwareprojekte abgebrochen wurden, ohne ein verwertbares Ergebnis geliefert zu haben. Im selben Jahr hatten 53% der Projekte bei Beendigung eine Budgetüberschreitung von über 50% zu verzeichnen. Mit dem RUP werden die Ursachen dieser Krise bekämpft, die in Faktoren wie unzureichendem Change Management, mangelhaften Tests und fehlendem Risikomanagement zu finden sind. Bedingung für das Gelingen ist, dass die Prozessrichtlinien diszipliniert eingehalten werden[75].

4.3 Prinzipien

Best Practices

ctices in der Terminologie des RUP sind "commercially proven approaches to software development that, n used in combination, strike at the root causes of software development problems."[76] Den folgenden Best ctices möchte der RUP folgen[77]:

1. Develop iteratively – Mit dem Ziel, allen kritischen Faktoren möglichst früh entgegenwirken zu können, werden kleine Releases – in der RUP-Terminologie Generations – erstellt. Im Rahmen dieser Releases erfolgt immer eine Testphase, welche sicherstellt, dass die kritischen Faktoren entdeckt werden.

Abbildung 13 gibt einen Überblick über den Ablauf eines iterativen Entwicklungsprozesses. Sie zeigt, wie die einzelnen Aktivitätsgruppen (z.B. Requirements) in jeder Iteration wieder durchlaufen werden. Diese Aktivitätsgruppen sind im RUP in Workflows definiert, auf die in Kapitel 4.4 detaillierter eingegangen wird.

[74] zur Softwarekrise vgl. Kruchten 2000, 4f.

[75] vgl. Versteegen 2000, 19.

[76] Kruchten 2000, 5.

[77] gemäss RUP 2000 eCoach, Abschnitt Best Practices.

Abbildung 13: Übersicht über einen iterativen Prozess[78]

[78] aus http://www.therationaledge.com/content/dec_00/f_rup.html (vgl. Anhang F).

2. `Manage Requirements` – Eines der Charakteristika von Anforderungen ist, dass sie sich verändern. Nicht nur lernt der Benutzer dazu, auch sein Arbeitsumfeld verändert sich. Dieser Umstand verlangt nach einem bewusst betriebenen Anforderungsmanagement. Dabei spielen `Use Cases` eine zentrale Rolle. Durch ihre visualisierende Komponente fördern sie das Erlangen eines gemeinsamen Verständisses der Geschäftsanforderungen.
3. `Model visually` – Durch den Einsatz von UML sollen Modelle erstellt werden, welche die Komplexität des Systems in einer übersichtlichen Weise und auf verschiedenen Abstraktionsniveaus überschaubar machen.
4. `Use component-based architectures` – Der RUP betont die Wichtigkeit von Architektur. Im Prozess wird darauf hingearbeitet, das System in einer Architektur zu bauen, die sich aus gekapselten und geschichteten Komponenten zusammensetzt.
5. `Verify quality` – Qualität wird im RUP wie folgt definiert:

   "- satisfies or exceeds an agreed upon set of requirements, and
   - assessed using agreed upon measures and criteria, and
   - produced using an agreed upon process."[79]

   Um diesem Qualitätsanspruch gerecht zu werden, sind Tests Bestandteile jedes Arbeitspaketes, jeder Iteration und jeder Phase. Sie werden ständig durch speziell damit beauftragte Personen durchgeführt.
6. `Control changes` – Eine der zentralen Herausforderungen in der Softwareentwicklung ist es, viele Entwickler in verschiedenen Teams, die möglicherweise an unterschiedlichen Orten gleichzeitig an unterschiedlichen Iterationen, Produkten oder Plattformen arbeiten, zu koordinieren. Eine solche Situation erfordert den disziplinierten Umgang mit Veränderungen an der Software. Im RUP existiert daher das Arbeitspaket `Configuration und Change Management`, welches diese Disziplin fördert.

Neben diesen sechs `Best Practices` realisiert der RUP noch 4 weitere, die allesamt für die Unterstützung der Projektarbeit mit den Werkzeugen von Rational werben:

- `Rational Tools` – Rational stellt eine Menge von Werkzeugen zur Verfügung, die in allen Phasen des Projektes Unterstützung bieten[80].
- `Tool-specific Guidance` – Die so genannten `Tool Mentors` stellen eine Verbindung zwischen dem RUP und anderen Werkzeugen von Rational her.
- `Process Guidance` – Benutzer von Rational Werkzeugen haben Zugriff auf kontextsensitive Hilfe, welche auf den RUP verweisen.
- `Configurable Process` – Der RUP stellt ein Framework zur Verfügung, mit dessen Hilfe der Prozess auf die individuellen Bedürfnisse einer Organisation zugeschnitten werden kann.

4.3.2 4+1 View

Für den Erfolg des Softwareentwicklungsprojektes ist es wichtig, dass alle Interessenvertreter – in der RUP-Terminologie die `Stakeholder` – ihre Anliegen einbringen können. Damit sich alle verstehen, sollen sie dies mit Hilfe von grafischen Modellen tun, die in der UML-Notation verfasst sind. So werden die Absichten durch übersichtliche Zeichnungen illustriert, die das gegenseitige Verständnis fördern.

[79] Gemäss RUP 2000 eCoach, Kapitel "Best Practices", Unterkapitel "Verify Quality", Abschnitt "Definition of quality".

[80] Darunter sind z.B.: Rational Rose als Entwicklungswerkzeug, Rational ClearCase als Konfigurationsmangementwerkzeug, Rational RequesitePro für das Anforderungsmanagement, Rational Suite TestStudio für das automatisierte Testen.

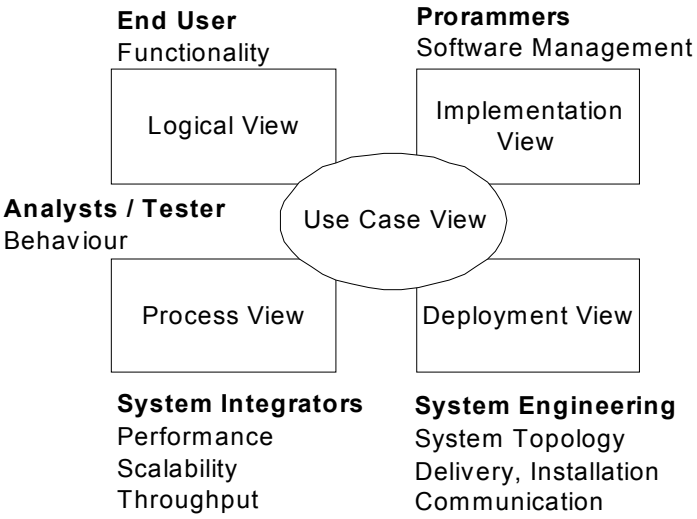


Abbildung 14: 4+1 View[81]

Die `4+1 View` (vgl. Abbildung 14) verdeutlicht, dass es bei der Abwicklung von Projekten fünf verschiedene Sichten auf das System gibt: Die Sicht der Analysten, die der Programmierer, die der Koordinatoren und die der System- und Netzwerkadministratoren. Diese Sichten werden von der im Zentrum stehenden `Use Case View`, der Sicht der Endbenutzer, zusammengehalten.

Die `Logical View` beleuchtet die logische Organisation des Systems. Sie reflektiert die Sicht der Analysten und der Designer.

Die `Implementation View` fokussiert auf die Funktionalität des Systems. Sie widerspiegelt die Sicht derjenigen, welche die `Use Cases` in Code umsetzen.

In der `Process View` werden die konkurrierenden Aspekte des Systems zur Laufzeit – Aufgaben oder Prozesse und deren Zusammenspiel – veranschaulicht.

Die `Deployment View` beschreibt die physische Verteilung der Software auf den unterschiedlichen Plattformen und zeigt die Sicht der System- und Netzwerkadministratoren.

Zu jeder View bietet UML Diagrammtypen, die der RUP zur Visualisierung der Sichtweise verwendet.

4.4 Prozessdefinition

In diesem Abschnitt wird beschrieben, wie sich der RUP präsentiert und welche Mittel den Prozess unterstützen. Dazu werden zuerst die Hauptprozesselemente `Worker`, `Activity` und `Artifact` erklärt, um deren Zusammenarbeit in `Workflows` verstehen zu können. Anschliessend werden die Elemente `Guidelines`, `Templates`, und `Tool Mentors` vorgestellt, welche zusätzliche Hilfestellung bei der Projektabwicklung bieten.

Im Prozess ist festgelegt, wer was wann wie tut. Dazu bedient er sich vierer Grundelemente, drei davon dargestellt in Abbildung 15:

[81] nach Kruchten 2000, 87.

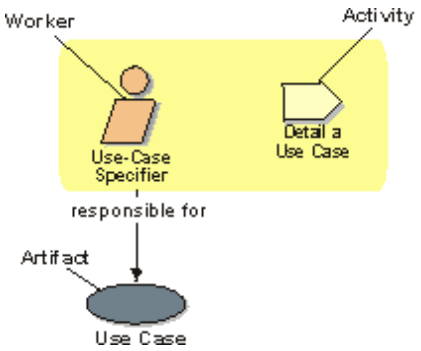


Abbildung 15: Die drei Grundelemente `Worker`, `Activity` und `Artifact`[82]

- `Workers` – das Wer: Rollen, welche die Teammitglieder einnehmen. Eine Person kann mehrere Rollen einnehmen und eine Rolle kann von mehreren Personen gleichzeitig wahrgenommen werden. Die Verteilung von Rollen auf Personen wird vom Projektmanager bei der Planung und Besetzung des Projektes vorgenommen. Der RUP definiert ungefähr 30 `Workers`.

  Beispiele: Designer, Use Case Author, Use Case Designer.

- `Activities` – das Wie: Eine `Activity` ist ein Arbeitsschritt, welcher von einer Person innerhalb einer Zeitdauer von ein paar Stunden und wenigen Tagen erledigt werden kann und der ein für das Projekt relevantes Ergebnis hervorbringt. `Activities` werden in `Activity Steps` unterteilt. Diese fallen in eine der Kategorien `Thinking Steps`, `Performing Steps`, `Reviewing Steps`.

  Beispiele: eine Iteration planen, `Use Cases` und `Actors` finden, das Design `reviewen`.

- `Artifacts` – das Was: `Activities` haben als Input und Output `Artifacts`. Ein `Artifact` ist ein Stück Information, das im Laufe eines Prozesses produziert, verändert oder verwendet wird und das dem Versionenmanagement unterliegt. Es sind die fassbaren Produkte des Projektes: Modelle, Dokumente, Quellcode, ausführbarer Code. Jedes `Artifact` ist in der Verantwortung eines einzelnen `Workers`, dem Eigentümer. Andere Personen können es gebrauchen und ändern, falls sie dafür die Erlaubnis erhalten.

  Abbildung 16 zeigt die zentralen `Artifacts` des RUP. Die Pfeile illustrieren den Informationsfluss zwischen ihnen.

  Zentrale Bedeutung haben das Visionsdokument und das Glossar. Das Visionsdokument enthält die Kernanforderungen an das System: Schlüsseleigenschaften, Einschränkungen und Rahmenbedingungen aus Benutzersicht. Es ist der Vertrag zwischen Kunden und Entwicklern.

  Im Glossar werden zentrale Begriffe der fachlichen sowie der technischen Welt definiert. Es dient dem gegenseitigen Verständnis und der einheitlichen Terminologie im Projekt.

  Zu fast allen `Artifacts` existieren `Guidelines`, welche als Anleitung zu ihrer Erstellung und zur Verwendung der in ihnen enthaltenen Informationen dienen.

[82] aus RUP 2000 eCoach, Kapitel Workers and Acitivities, Workers Overview.

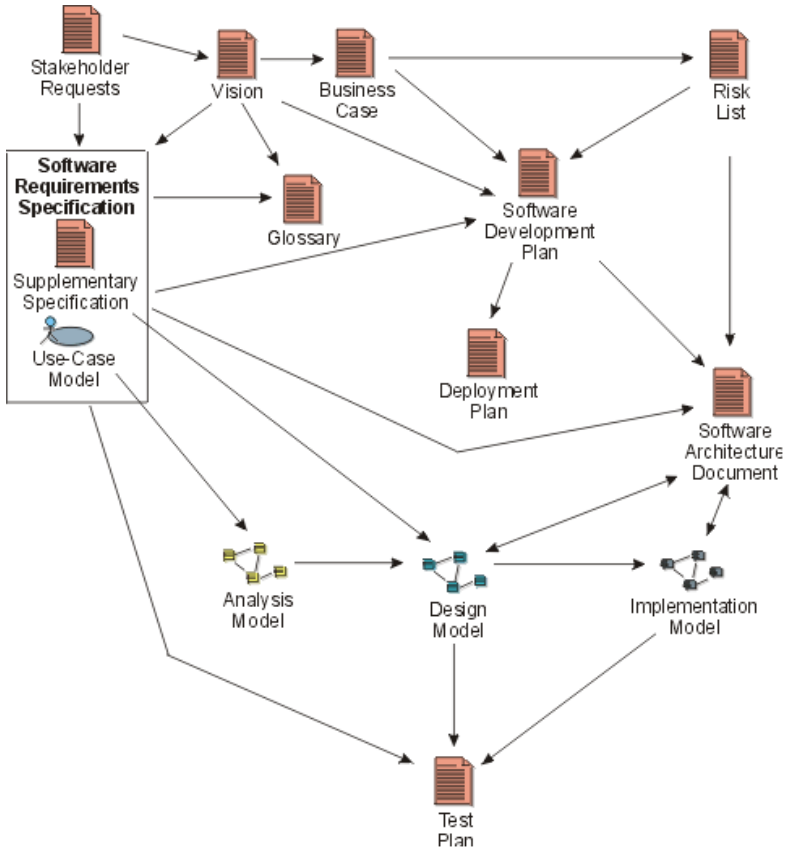


Abbildung 16: Zentrale Artifacts in RUP und der Informationsfluss zwischen ihnen[83]

- `Workflows` – Der RUP besteht aus 9 `Workflows`[84]. Jeder `Workflow` zeigt, wer was wann wie zu tun hat, und wird in den Abschnitten `Introduction`, `Concepts`, `Workflow Details`, `Activity Overview`, `Artifact Overview` und `Guidelines Overview` beschrieben. Alle Inhalte sind mit Zeichnungen, die sich an eine einprägsame Symbolik halten, visualisiert. Ein Beispiel für die Visualisierung ist Abbildung 17, welche den `Requirements Workflow` illustriert. Die Elemente der Zeichnungen sind in RUP 2000, eCoach, verlinkt, so dass man beispielsweise von der Übersicht der `Artifacts` auf ein einzelnes `Artifact` klickend dessen Beschreibung anzeigen kann. Diese enthält wiederum Hinweise darauf, welche anderen `Workflows` dieses `Artifact` benutzen – als In- oder als Output. Zu diesen weiteren `Workflows` kann man dann wieder per Mausklick navigieren.

[83] aus RUP 2000 eCoach, Kapitel Artifacts.

[84] vgl. , Seite 26, vertikale Achse.

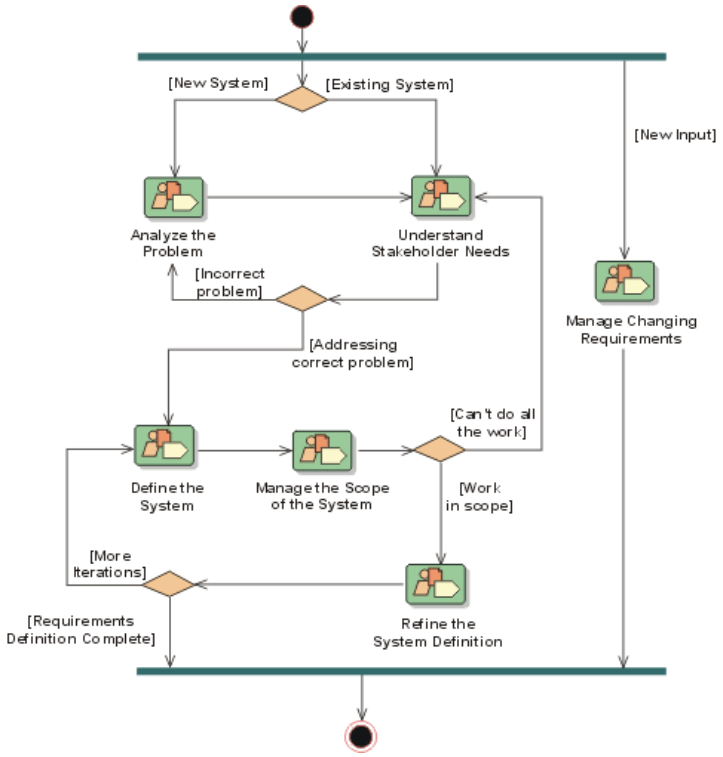


Abbildung 17: Requirements Workflow[85]

Neben den vier Grundelementen werden die folgenden drei weiteren Prozesselemente angeboten, welche die Projektführung erleichtern sollen:

- `Guidelines`: `Guidelines` sind Regeln, Empfehlungen oder Heuristiken, die bei der Umsetzung von `Activities` helfen. Es existieren `Guidelines` zu den Themen `Workshops, Requirements Elicitation` und `Review.`

  Beispiel: `Guidelines` zum Führen von Interviews.

- `Templates`: RUP stellt eine Reihe von Vorlagen für `Artifacts` zur Verfügung. Sie liegen in Microsoft Word, Adobe FrameMaker, HTML und Microsoft Project vor.

  Beispiel: Wordvorlage für die Beschreibung der Architektur.

- `Tool Mentors`: Viele `Activities` im RUP werden von Rational Software-Werkzeugen unterstützt. Die `Tool Mentors` enthalten detaillierte Beschreibungen dazu, wie diese Werkzeuge einzusetzen sind.

  Beispiel: Anleitung zur benutzergerechten Veränderung des Rational Unified Process.

Diese vorgestellten Grundelemente repräsentieren die Prozessstruktur. Die dynamische Struktur wird durch das Anwenden des Prinzips der iterativen Vorgehensweise in den einzelnen Phasen gelebt.

[85] aus RUP 2000 eCoach, Kapitel Core Workflows, Requirements.

5 Beurteilung

5.1 Vorgehen und Konventionen

Nachdem wir uns in den beiden vergangenen Kapiteln ein grobes Bild der beiden Softwareprozesse RUP und XP gemacht haben, können wir nun dazu übergehen, eine Beurteilung vorzunehmen. In den folgenden Abschnitten werden dazu die Antworten der beiden Prozessmodelle XP und RUP auf die Herausforderungen des Wissensmanagements untersucht. Als Grundlage für die Beurteilung dienen die Publikationen der Verfechter der Prozesse sowie im Falle von RUP die Online-Dokumentation von Rational. Die Beurteilung schliesst daher nur diejenigen Eigenschaften der Prozesse ein, die in diesen Medien dokumentiert sind. Wurde in der Dokumentation eines Prozesses zu einer Herausforderung keine Aussage gefunden, so wird der Prozess diesbezüglich als neutral eingestuft.

Der Aufbau der Kapitel 5.3 und 5.4, die sich mit der Beurteilung der einzelnen Prozessschritte[86] befassen, folgt folgendem Schema:

1. Tabellarische Darstellung der Antworten auf die Herausforderungen des jeweiligen Schrittes. Dabei kommen die folgenden Bewertungen zum Zug:

   Strahlendes Gesicht: Der Softwareprozess bietet gute Möglichkeiten zur Bewältigung dieser Herausforderung.

   Neutrales Gesicht: Der Softwareprozess lässt die Bewältigung dieser Herausforderung offen.

   Trauriges Gesicht: Der Softwareprozess hat Merkmale, welche der Herausforderung widersprechen.

   In den Tabellen sind diejenigen Herausforderungen hellgelb hinterlegt, deren Zutreffen von den Beurteilenden als schädlich eingestuft wurde. Ein entsprechender Kommentar findet sich vor oder nach der jeweilige Tabelle.

   Die Tabelle wird durch eine zusammenfassende Zeile abgeschlossen. Dort werden die Anzahl strahlender, neutraler und trauriger Gesichter gezählt und eine Gesamtbewertung vorgenommen. In dieser Gesamtbewertung hebt ein trauriges Gesicht ein strahlendes Gesicht auf. Ein neutrales Gesicht hat keinen Einfluss auf die Bewertung.
2. Kurze Zusammenfassung der Ergebnisse.
3. Tabellarische Zusammenfassung der Ergebnisse.

Die Beurteilung wurde von mir und 4 Experten der Softwareentwicklungsprozesse vorgenommen:

(Die beiden RUP-Experten haben nur RUP und die beiden XP-Experten nur XP mit "smileys" beurteilt. Frau Stucki hat sowohl zu RUP als auch zu XP eine Beurteiluing gemacht. Deshalb gibt es in der Spalte für RUP und in der Spalte für XP je drei "smileys" (Anm. H.-P. Korn, 7. Jan. 2004)

RUP	XP
Fiorenzo Maletta RUP-Experte und Projektleiter	*Ulrike Adrion* Prozessexpertin im Competence Center OO
Roland RUP-Experte und Coach	*Anton Liebetrau* Leiter Competence Center OO

Bevor aber die Resultate der Beurteilung vorgestellt werden, wird in einem ersten Abschnitt dargestellt, wie die Prozessmodelle die Voraussetzungen schaffen, welche die Entwicklung

[86] Bei der Durchsicht mag es sich als nützlich erweisen, Abbildung 3, Seite 10, und Abbildung 4, Seite 11, zu Rate zu ziehen.

und das Nutzen von Wissen überhaupt erst ermöglichen[87]. Dies beantwortet die Frage, welche Kultur der Zusammenarbeit aus dem Einsatz der Prozesse resultiert.

Nach der Beurteilung wird ein Fazit gezogen und Empfehlungen zum Einsatz der beiden Prozessmodelle abgegeben.

5.2 Voraussetzungen schaffen

5.2.1 Extreme Programming

XP möchte sich im Projektablauf auf das Wichtige und Notwendige konzentrieren. Damit gehört der Ansatz zu den in den 90er Jahren entstandenen so genannten leichtgewichtigen Prozessen. Im Unterschied zu den konventionellen, schwergewichtigen Prozessen sind in einem leichtgewichtigen Prozess nur wenige, einfach zu befolgende Regeln und Praktiken notwendig. Dokumente werden gar nicht oder nur wenige erstellt. Der Entwickler ist frei von Verfahrensanweisungen und kann kreativ und produktiv arbeiten. Er konzentriert sich auf das Ziel seines Tuns[88]. XP legt grossen Wert auf die Kreativität, Autonomie und Individualität der Mitarbeiter und unterstützt damit eine Unternehmenskultur, die eine Voraussetzung für die Nutzung von Wissen überhaupt ist. Die Forderungen des `Agile Manifesto`, dem XP verschworen ist, betonen die Wichtigkeit dieses Ansatzes:

"- Individuals and interactions over processes and tools
- working software over comprehensive documentation
- customer collaboration over contract negotiation
- responding to changes over following a plan"[89]

und:

"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."[90]

Als wenig geregelter Prozess lässt XP den Projektmitgliedern viel Freiraum. Sie können Neues ausprobieren und sich als Team jederzeit so organisieren, wie es optimal erscheint. Diese Freiheit spielt dem so genannten Hawthorne-Effekt in die Hände, der – etwas salopp ausgedrückt – besagt, "dass Mitarbeiter besser arbeiten, wenn sie etwas Neues ausprobieren"[91] und impliziert, dass nicht standardisiertes Vorgehen die Regel sein sollte.

Kreativ arbeiten ist nur möglich, wenn Geist und Körper ausgeruht sind[92]. Um dies sicherzustellen, betont XP die Notwendigkeit, nicht mehr als 40 Stunden pro Woche zu arbeiten.

In XP gilt der Grundsatz, dass bei Interessenkonflikten von den vier Variablen Kosten, Termine, Qualität und Umfang die Qualität die Unveränderliche sein sollte. Im Vordergrund stehen dabei die Ansprüche, die der Mensch an das Produkt seiner Arbeit stellt:

"Everybody wants to do a good job, and they work much better if they feel they are doing good work. If you deliberately downgrade quality, your team might go faster at first, but soon the demoralization of producing crap will overwhelm any gains you temporarily made from not testing, or not reviewing, or not sticking to standards."[93]

---

[87] vgl. dazu Kapitel 2.1.1, Seite 7.

[88] vgl. http://www.extremeprogramming.org/light1.html und http://www.extremeprogramming.org/light2.html (Anhang G).

[89] Martin 2001, 8.

[90] Martin 2001, 13.

[91] De Marco et al. 1998, 139.

[92] De Marco et al. 1998, 24. In diesem Punkt gibt es auch andere Auffassungen: Es existiert die Meinung, dass hohe Anforderungen gestellt und sozialer Druck aufgebaut werden sollte, um das Potential in den Köpfen von Mitarbeitern mit Hochschulabschluss gewinnbringend zu nutzen. "Die legendären 80-Stunden-Wochen ... dienen einem ernstzunehmenden Weiterbildungszweck: Sie befähigen die Besten, sich entlang einer Lernkurve nach oben zu bewegen... ." (North 2002, 129) Diese Meinung wird in dieser Arbeit nicht geteilt, da wir der Überzeugung sind, dass eine solche Nutzung – mit dem Vokabular von Umweltwissenschaftlern – nicht nachhaltig ist.

[93] Beck 2000, 18.

Damit ist XP in Übereinstimmung mit einem weiteren Element der Unternehmenskultur, das Voraussetzung ist für die optimale Nutzung der Ressource Wissen im Softwareprozess. Diese Auffassung wird weiter unterstützt durch das starke Gewicht, das XP auf die Arbeit im Team und auf Gespräche legt: "Qualitätskult ist der stärkste Katalysator zur Teambildung. Es bindet das Team zusammen, weil es sich vom Rest der Welt unterscheidet."[94]

5.2.2 Rational Unified Process

Wie vor allem aus den Schilderungen zur Prozessdefinition (Kapitel 4.4, Seite 30) ersichtlich, bietet der RUP extensive Anleitungen zur Projektabwicklung. Sehr genau ist beschrieben, wer wann was zu tun hat. Dies hilft dem Projektteam, Fehler und Auslassungen zu vermeiden. Ebenfalls sind dadurch einheitliche Dokumentation, Managementkontrolle und Standardisierung gewährleistet.

Doch hat diese Genauigkeit auch ihre Nachteile. Es besteht die Gefahr, dass die Arbeit in ein Korsett gezwängt wird, dass ein Wust von Papier entsteht und dass Verantwortungsbewusstsein verloren geht[95]. Die Folge davon ist ein Einbruch der Motivation. Will man diesen Gefahren entgehen, muss die `Best Practice "Manage Process"` umgesetzt werden, so dass der Prozess genau auf die Bedürfnisse des Projektteams zugeschnitten ist.

Bezüglich der Codierung bietet der RUP anders als XP wenig Unterstützung. Hier kommt vor allem der `Tool Mentor` zum Zuge, der auf die Produkte von Rational oder Microcsoft verweist. So wiegt die Tatsache schwer, dass es sich beim RUP um ein kommerzielles Produkt handelt[96]: Für jede denkbare Tätigkeit wird ein unterstützendes Werkzeug von einem der beiden Unternehmen angeboten. Diese Toolverlinkung bremst den Einsatz des RUP in Unternehmen, die diese Produkte nicht kaufen wollen oder können. Auch ist fragwürdig, ob die enge Verknüpfung zwischen Prozess und Werkzeugen, die ja gerade als Stärke des RUP hervorgehoben wird, in einem Vorgehensmodell angebracht ist.

Beim RUP handelt es sich um einen schwergewichtigen Prozess, der die gesamte organisatorische Palette der Projektabwicklung abdeckt und dabei viel Gewicht auf Dokumentation legt[97]. Bei der Einführung empfiehlt es sich, die Projektmitglieder in der Handhabung des Prozesses wie auch der Werkzeuge zu schulen[98].

Die Grundidee des RUP – mittels visualisierenden Modellen die Komplexität überschaubar zu machen und die Kommunikation zu vereinfachen – ist begeisternd. Allerdings wird die Begeisterung relativiert durch den eben besprochenen extensiven Tooleinsatz und der starken Formalisierung. Die Gefahr, sich im Dickicht von Dokumenten und Werkzeugen zu verlieren, ist nicht zu unterschätzen.

Als weitere Voraussetzung der Wissensentwicklung haben wir die Freiheit der Entwickler identifiziert, qualitativ hochstehende Arbeit abliefern zu können. Auch im RUP steht die Softwarequalität im Vordergrund[99]. Ein System ist dabei dann von guter Qualität, wenn es tut, was es sollte. Es ist also notwendig, dass zwischen den `Stakeholdern` bei der Definition der Anforderungen ein gemeinsames Verständnis darüber erreicht wird, was das System tun sollte und wie die Funktionalität getestet wird. Durch den `Workflow` des Anforderungsmanagements wird dies gewährleistet. Wie in XP gibt es auch im RUP keine dedizierte Rolle, die sich mit Qualität beschäftigt. Vielmehr ist auch hier Qualität in der

---

[94] De Marco et al. 1998, 176.

[95] vgl. De Marco et al. 1998, 134-136.

[96] Der RUP ist zur Zeit (25. Mai 2003) bei http://www.rational.com/products/rup/index.jsp für 695$ zu erwerben.

[97] Kruchten 1996 betont im letzten Abschnitt, dass RUP keinen grossen Fokus auf Dokumente legt. Angesichts der über 80`Artifacts`, möchten wir dieser Aussage widersprechen.

[98] RUP 2000 eCoach, unter Core Workflows, Environment, Concepts, Implementing a Process in an Organization, Plan Process Implementation, Plan Training.

[99] vgl. Kruchten 2000, 12f und 25f.

Verantwortung aller. Sie wird allerdings von dedizierten Rollen mittels Tests und Metriken überprüft.

5.3 Die taktischen Schritte

5.3.1 Beschaffen

Die Bedeutung des Prozessschrittes Beschaffen wurde in der Diagnose eher gering eingestuft.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang
Beschaffen	6	5	4	5

Tabelle 4: Resultate der Diagnose für den Prozessschritt Beschaffen

Wie Tabelle 5 zeigt, haben beide Prozesse vielfältige Antworten auf die Herausforderungen dieses Prozessschrittes. Es zeigen sich hier die unterschiedlichen Schwerpunkte der beiden Prozessmodelle: Während der RUP eher dokumentenorientiert operiert und die Informationen daher primär mittels IT-Unterstützung beschafft werden, verlässt sich XP auf die Kommunikation unter den Teammitgliedern. Dies bedeutet, dass die Teammitglieder enge Beziehungen pflegen und damit die weichen Faktoren der Persönlichkeit von grosser Bedeutung sind.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Artikulation		
Machen sie sich klar, wofür sie die Information brauchen und drücken Sie das deutlich aus.	Der Prozess begründet jedes `Artifact`, bindet es in einem `Workflow` ein und teilt ihm einen Verantwortlichen sowie einen Adressaten zu. Damit ist klar, wofür die Informationen, die in den `Artifacts` festgehalten werden, zu gebrauchen sind. 	Informationen werden aus dem Code, den Tests und von anderen Teammitgliedern beschafft. Dies verlangt, dass die Informationssuchenden sich klar darüber sind, wofür sie die Informationen brauchen. Im Planning Game muss der Nutzen einer Anforderung dargelegt werden. Dies fördert, dass nachgefragt wird.
Wenden Sie sich mit Informationswünschen an die richtige Adresse.	Damit, dass der Prozess die `Artifacts` verantwortlichen Personen zuordnet, fördert er, dass der Informationssuchende sich an die richtige Person wendet. Durch die Rollendefinition besteht die Gefahr, dass niemand sich zuständig fühlt. 	Im `Stand-Up Meeting` werden Fragen an die Runde gestellt. So findet man denjenigen, der weiterhelfen kann. Wenige Rollen und kein Teilen von Verantwortung.
Bewusstsein		
Schaffen Sie Wegweiser: Verzeichnisse, Gelbe Seiten, Karten.	Der eCoach ist der Wegweiser, falls die Rollenzuteilung bekannt ist. Die Rollenzuteilung würde die Aufrechterhaltung von Wissenlandkarten ermöglichen Es wird empfohlen, Checklisten zu Projektbeginn oder nach jeder Iteration zu erstellen, die Aufschluss über die Skills im Team geben sollen und den weiteren Einsatz der Mitglieder planen lässt[100]. 	XP ist für kleine Teams konzipiert. Skilllisten werden keine geführt.
Nutzen Sie Interessengemeinschaften, um Wissen innerhalb der Organisation offen zu legen.	Sowohl der RUP wie auch XP erfreuen sich einer grossen Fangemeinde. Die zahlreichen Websites zeigen, dass die Mitglieder der Fangemeinde sich gerne austauschen. Dies gilt als einer der grössten Vorteile der objektorientierten Welt überhaupt und wirkt sich auch auf den Wissensaustausch innerhalb des Unternehmens positiv aus. Es wird zwar ein Respository geführt, auf das jeder Zugriff hat, der RUP macht aber explizit zu Interessengemeinschaften keine Aussage. 	Sowohl der RUP wie auch XP erfreuen sich einer grossen Fangemeinde. Die zahlreichen Websites zeigen, dass die Mitglieder der Fangemeinde sich gerne austauschen. Dies gilt als einer der grössten Vorteile der objektorientierten Welt überhaupt und wirkt sich auch auf den Wissensaustausch innerhalb des Unternehmens positiv aus. XP kennt keine Grenzen der Kommunikation.
Zugang		

[100] vgl. Versteegen 2000, 166.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Schaffen Sie ein sinnvolles Gleichgewicht zwischen aktivem Suchen und Information, die Ihnen aufgedrängt wird.	Der Prozess regelt klar, welche Dokumente wem zugestellt werden sollen. Es wird nicht über den Wert der freiwilligen Informationssuche gesprochen. Der RUP geht von aktivem Suchen aus, es gibt keine `push` Informationen. Das ausgeprägt iterative Vorgehen bewirkt, dass sich Dokumente oft ändern. Es besteht also die Gefahr, dass die Informationen nicht bis zu den Teammitgliedern durchdringen. Das Produkt bietet mit dem eCoach jedem die Möglichkeit, sich über den Prozess selbst zu informieren. Dokumente entstehen durch Externalisierung. Ihr Inhalt ist daher einfacher zugänglich als das Wissen, das nur in den Köpfen der Mitarbeiter vorhanden ist. 	Die Informationsbeschaffung erfolgt im direkten Gespräch. Sowohl Push wie auch Pull verlässt sich auf die Kommunikation im Team.
Um die Werkzeuge zum Suchen und Finden bedarfsgerecht zu gestalten, muss der Nutzer involviert werden.	Der eCoach wurde aus Projekterfahrung erstellt. Seine Mächtigkeit birgt allerdings die Gefahr, sich in ihm zu verlieren. Die grosse Anzahl der eingesetzten Werkzeuge macht einen grossen Erfahrungsschatz zur Voraussetzung. Die Entwicklungswerkzeuge bieten umfangreiche Debug- und Testfunktionalitäten, die dem für den Entwickler optimal ausgerichtet sind und kontinuierlich weiterentwickelt werden 	Das Werkzeug zum Suchen und Finden ist die Kommunikation sowie das eingesetzte Entwicklungswerkzeug. Es gibt eine sehr beschränkte Anzahl Werkzeuge. Die Entwicklungswerkzeuge bieten umfangreiche Debug- und Testfunktionalitäten, die dem für den Entwickler optimal ausgerichtet sind und kontinuierlich weiterentwickelt werden.
Anleitung		
Bibliothekare werden zu Internet-Experten.	Der RUP macht hierzu keine Aussage. 	XP macht hierzu keine Aussage.
Schaffen sie eine neue Stelle: den Wissensmanager.	Der RUP macht hierzu keine Aussage. 	XP macht hierzu keine Aussage.
Experten fungieren als Informationsfilter.	Es gibt viele Spezialistenrollen. Sie geben in das Projektteam nur diejenigen Informationen weiter, die wichtig sind. 	Es gibt keine Filterfunktionen.
Vollständigkeit		
Ermöglichen Sie einen leichten Zugriff auf zentral und dezentral verwaltete Informationen.	Der Zugriff auf die Informationen ist durch die Definition der `Workflows` gewährleistet. Es ist alles zentral verwaltet. 	Code und Tests sind vollumfänglich verfügbar. Es wird alles zentral verwaltet.
Schaffen Sie die nötigen Rahmenbedingungen und Prozesse, um die Wiederverwendung von Wissen zu fördern.	`Best Practices` sind integriert im Prozess und sind als wiederverwendetes Wissen zu betrachten. Dank der Offenheit des Prozesses können weitere `Best Practices`, die aus der eigenen Erfahrung gewonnen wurden, integriert werden. 	Die Tests repräsentieren das Wissen über die Korrektheit eines Stückes Code. Dieses Wissen wird immer wieder verwendet. Mit der Praktik des `Refactorings` wird das Wissen über das Erstellen von einfachem Design und Code wiederverwendet.
Zusammenfassung Einschätzung	22x 10x 1x	30x 2x 1x
Ergebnis Zusammenzug	21x	39x

Tabelle 5: Gegenüberstellung von XP und RUP für den Prozessschritt Beschaffen

Die gelb markierten Zeilen in Tabelle 5 beinhalten die Herausforderungen, die bei der Diagnose konträr beurteilt wurden. Experten- und Spezialistenrollen werden im Softwareprozess nicht geschätzt. XP schneidet daher in dieser Hinsicht besser ab als der RUP. Ebenfalls ist es im Softwareprozess von Vorteil, wenn alle Informationen zentral verwaltet werden und für alle zugänglich sind. Beide Prozessmodelle entsprechen diesem Wunsch.

Einen Neueinsteiger wird das RUP-Team zuerst mit dem Erlernen der eingesetzten Werkzeuge sowie dem Studium der `Artifacts` beauftragen, während er in XP seine

Informationen bei den anderen Teammitgliedern, aus dem Code und den Tests beschaffen muss. Die Vielfältigkeit der Informationsbeschaffung ist damit in XP geringer, was es erleichtert, den Überblick zu behalten. Allerdings hat die starke Personenorientierung den Nachteil, dass Informationen flüchtig sind: Ist die Person abwesend, so ist die Information nicht erreichbar.

Bei Redimensionierungen leidet ein XP-Team mehr als ein RUP-Team, denn die Personen, von denen das Wissen beschafft werden kann, werden vielleicht nicht mehr verfügbar sein. Der Umstand, dass im RUP viel mehr externalisierte Informationen vorhanden sind, bietet den Vorteil, dass das Beschaffen viel unabhängiger von entlassenen Personen ist.

Tabelle 6 fasst die Resultate der Gegenüberstellung zusammen.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang	RUP	XP
Beschaffen	6	5	4	5	21x	29x

Tabelle 6: Resultate der Gegenüberstellung für den Prozessschritt Beschaffen

5.3.2 Nutzen

Die Diagnose hat gezeigt, dass dem Schritt Nutzen im Softwareprozess grosse Bedeutung beigemessen wird, während ihm im Zusammenhang mit Entlassungen und Einstellung weniger Gewicht zukommt:

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang
Nutzen	3	4	5	4

Tabelle 7: Resultate der Diagnose für den Prozessschritt Nutzen

Softwareentwicklung hat zum Ziel, Prozess- und Produktwissen in Software abzubilden, so dass der Kunde die Software zur Abwicklung seiner Arbeit einsetzen kann. Die Softwareentwickler müssen also das Wissen um die Prozesse und die Produkte mit ihrem technischen Know-how verbinden. Dafür sind Durchlässigkeit und Freiheit notwendig: Durchlässigkeit erlaubt es, Informationen fliessen zu lassen, so dass sie am richtigen Ort genutzt werden können. Freiheit ist notwendig, um beispielsweise eine Technologie auszuprobieren, über die man von jemand anderem erfahren hat.

Tabelle 8 zeigt, wie RUP und XP den Herausforderungen dieses Schrittes des Wissensmanagements begegnen.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Durchlässigkeit		
Verändern Sie die Organisationsstruktur, um die Kommunikation und den Wissensfluss zu verbessern.	Kommunikation und Wissensfluss sind in `Workflows` geregelt. Starre Rollenzuteilung baut tendenziell Barrieren auf. Die Rollen werden aber explizit dafür definiert, damit sie demjenigen zugeteilt werden können, der dafür am besten geeignet ist, so dass eine optimale Organisationsstruktur aufgebaut werden kann und der optimale Fluss der Artefakte gewährleistet ist. 	Das sich selbst organisierende Team strebt jederzeit danach, sich so zu strukturieren, dass optimal zusammengearbeitet werden kann. `Customer on site` und `Planning Game` sorgen für die maximale Durchlässigkeit zwischen Entwicklern und Kunden.
Entwerfen Sie ein physisches Umfeld, das zur gegenseitigen Befruchtung von Ideen beiträgt.	Der RUP macht hierzu keine Aussagen. 	Das Bürolayout muss so gestaltet werden, dass `Pair Programming` möglich ist, dass jeder aber auch noch etwas persönlichen Platz hat.
Behandeln Sie Informationen wie eine sprudelnde Quelle, die sich überall in der Organisation ausbreiten kann.	Der Informationsfluss ist klar geregelt. Von einer sprudelnden Quelle kann daher nicht gesprochen werden. 	Im Entwicklungsprozess kann dank dem anwesenden Kunden, den kontinuierlichen `Stand-up-Meetings` und dem `Pair Programming` der Informationsfluss als sprudelnd betrachtet werden. Für das Unternehmen allerdings ist die starke Teamorientierung nachteilig.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Arbeiten Sie routinemässig mit allen Gemeinschaften zusammen, die an der Information einen Anteil haben.	Zusammenarbeit mit den Kunden und Endbenutzern und deren Bedeutung als Informationsquellen wird zwar betont[101], diese werden aber immer als ausserhalb des Projektes stehend betrachtet. Die Anforderungsanalyse wird nicht primär von einem Kunden vorgenommen, sondern es existiert ein `Analyst`, der sogar für die Priorisierung der `Change Requests` zuständig ist.	Durch die Praktik des `Customer on site` sind Kunde und IT-Team gleichermassen an der Arbeit beteiligt. Sie sind räumlich nicht getrennt, was die Zusammenarbeit zu intensivieren vermag. Die Rechte und Pflichten von Kunde und Technikern sind vereinbart.
Freiheit		
Würdigen Sie die Beiträge von allen Leuten aus dem Unternehmen.	Jede Rolle ist verantwortlich für gewisse `Artifacts`. Diese wird er auch kontrollieren und versionieren. Somit ist es möglich, die Beiträge des Einzelnen zu würdigen.	`Collective Code Ownership` impliziert, dass es nicht von herausragender Bedeutung ist zu wissen, wer was getan hat. Das ganze Team ist für jeden Teil des Produktes verantwortlich. Dies verunmöglicht es, die Arbeit eines Einzelnen hervorzuheben.
Lassen Sie Zeit und Raum für Spiele.	Prototypen sind Elemente des Prozesses. Der Prozess ist aber nicht auf Entspannungsphasen ausgelegt.	Zum Ausprobieren werden `Spikes` verfasst. Planning Game; Klares Commitement dazu, dass Arbeit Spass machen muss.
Zusammenfassung Einschätzung	 10 x 	16x
Ergebnis Zusammenzug		14x

Tabelle 8: Gegenüberstellung von XP und RUP für den Prozessschritt Nutzen

Während im RUP der Fluss von Informationen über die `Workflows` gesteuert wird, folgt der Informationsfluss in XP keinen Regeln: Der Kunde ist mit den Entwicklern in demselben Raum. So können Gespräche dann stattfinden, wenn sie notwendig sind.

In XP wird der Kunde zum Team gezählt. Im RUP dagegen werden weder Kunden noch Endbenutzer als Teil des Projektteams betrachtet. Dies ist dem Wissensfluss weniger zuträglich. Die strenge Formalisierung limitiert unter Umständen die Kreativität der Informationssuchenden bei der Suche zusätzlich: Sie verlassen sich auf die Ratschläge des Prozesses statt selbst auf die Suche zu gehen und verpassen damit unter Umständen wertvolle Beiträge. Des Weiteren schafft die klare Rollenzuteilung des RUP Zuständigkeitsgrenzen, welche der Durchlässigkeit abträglich sein können.

Der RUP bietet allerdings dank der klaren Regelung von Verantwortlichkeiten den Vorteil, dass jederzeit klar ist, wer für einen bestimmten Beitrag zeichnet, während sich in einem XP-Projekt alle Teammitglieder für alle Produkte gleichermassen verantwortlich zu fühlen haben. So bietet der RUP immerhin die Chance, die Beiträge zu würdigen.

Beide Prozesse integrieren die Möglichkeit des Ausprobierens. Während der RUP mit Prototypen arbeitet, werden in XP `Spikes` geschrieben.

Tabelle 9 stellt die Resultate der Beurteilung zusammen.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang	RUP	XP
Nutzen	3	4	5	4		14x

Tabelle 9: Resultate der Gegenüberstellung für den Prozessschritt Nutzen

[101] vgl. u.a. Kruchten 2000, 167.

5.3.3 Lernen

Die Diagnose hat gezeigt, dass Lernen als wichtigstes Element für den erfolgreichen Softwareprozess betrachtet wird. Im Rahmen der Einarbeitung neuer Mitarbeiter sowie in der Situation von Entlassungen ist es ebenfalls von zentraler Bedeutung (vgl. Tabelle 10).

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang
Lernen	1	3	3	2

Tabelle 10: Resultate der Diagnose für den Prozessschritt Lernen

Lernen – mehr als alle anderen Schritte des Wissensmanagements – ist nur beschränkt direkt steuerbar. Hier bedarf es vor allem eines Arbeitsumfeldes, welches das Lernen fördert[102], denn Lernen ist nicht immer ein bewusster Vorgang. Häufig lernt man durch unmittelbare Erfahrung und durch Versuch und Irrtum. Daher muss ein Softwareprozess `Learning on the job` fördern. Der Umgang mit Fehlern muss von Offenheit geprägt sein, damit aus ihnen gelernt werden kann.

Tabelle 11 stellt XP dem RUP gegenüber und gibt Hinweise auf die Art und Weise, wie sie mit den Herausforderungen des Schrittes Lernen umgehen.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Sichtbarkeit		
Stellen Sie die Komplexität menschlicher Systeme dar.	Die Rollen und ihre Funktionen sind sehr ausführlich geregelt. In genauen Beschreibungen von `Workflows` werden Interaktionen geregelt. Die Komplexität soll so überschaubar gemacht werden. Dies geschieht mit umfangreichen visuellen Hilfsmitteln. 	Wenige Rollen werden definiert. Der Komplexität wird mit der Betonung auf die notwendige Kommunikation begegnet. Was den Erfolg des Teams ausmacht, ist seine Fähigkeit, informelle Netzwerke zu nutzen[103]. Die informelle Komponente wird in XP gepflegt und betont.
Beziehen Sie Körper und Geist zusammen ein, um „weiche“ Aspekte mit der Strategie zu verbinden.	Der RUP macht diesbezüglich keine Aussagen. Die starke Formalisierung lässt die Tendenz zur Vernachlässigung dieser Komponente erkennen. 	40-Stunden Woche: Ausgeruhter und damit kreativer Geist in ausgeruhtem Körper. Immer wieder wird betont, dass die Teammitglieder eines extremen Teams sich durch bestimmte persönliche Fähigkeiten wie Kreativität, Freude an der Zusammenarbeit und Spass an Kommunikation auszeichnen müssen.
Gewohnheit		
Betonen Sie bei der Arbeit den Vergnügungsaspekt.	Strenge Formalisierung hemmt spassige Kreativität. 	Dass Arbeit Spass machen soll, betont Beck immer wieder: "XP is a lightweight, efficient, low-risk, predictable, scientific and fun way to develop software."[104] Seine Publikationen sind mit witzigen Anekdoten gespickt und bringen den Leser zum Schmunzeln.
Sorgen Sie dafür, dass Mechanismen, die zum Nachdenken anregen, in die tägliche Arbeit eingebaut sind.	Kruchten[105] betont, dass dem iterativen Ansatz ständiges Entdecken, Lernen und Implementieren inhärent ist. Das Team verfolgt in den Iterationen das Ziel, die `Artifacts` bis zum Ende der Phase zum Abschluss zu bringen, und zwar in einer nachvollziehbaren und wiederholbaren Art. Am Ende der Phase wird über die Ergebnisse nachgedacht, so dass immer über den Stand des Projektes nachgedacht wird. Formale Prozesse verhindern aber zumindest tw. die Fähigkeit eines Systems, zu lernen und sich selbst zu heilen[106]. 	Während im `Pair Programming` der Driver programmiert, denkt der Partner nach[107]. In den Praktiken `Refactoring` und `Simple Design` ist jeder ständig aufgefordert, einfachere Lösungen zu suchen.

[102] vgl. Kapitel 2.1.1, Seite 7.

[103] vgl. dazu auch Goleman 1996, 206f.

[104] Beck 2000, XVII.

[105] Kruchten 2000, 7.

[106] vgl. De Marco et al. 1998, 13.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Nutzen Sie Fehler, Missverständnisse und Misserfolge aus, um etwas daraus zu lernen.	Der `Change Management Workflow` regelt den Umgang mit Missverständnissen. Dass Missverständnisse früh im Prozess erkannt werden und dass aus Fehlern kontinuierlich gelernt werden kann, wird als Vorteil des iterativen Vorgehens betont[108]. Durch die Betonung des Kostenfaktors von Fehlern wird der freie Umgang aber gehemmt, da niemand gerne zugibt, früher einen Fehler begangen zu haben, der sich heute als sehr teuer herausstellt. 	Es gilt der Grundatz `Embrace Change`. Fehler werden mit Kosten nicht mehr direkt in Verbindung gebracht. Dies erlaubt einen ungezwungeneren Umgang mit Fehlern.
Fördern Sie die Kunst des `Learning by doing`.	Als Vorteil beim Einsatz von RUP wird hervorgestrichen, dass der hohe Standardisierungsgrad und die `Artifacts` das Einarbeiten eines neues Mitarbeiters erleichtern, da er mit Hilfe des `eCoaches` sofort beginnen kann zu arbeiten. `Reviews` sind Bestandteil des Prozesses. 	`Code Reviews` werden ersetzt durch `Pair Programming`[109]. Neue Mitarbeiter werden in den Prozess integriert und arbeiten mit einem Partner zusammen.
Zusammenfassung Einschätzung	4x 9x 5x	18x
Ergebnis Zusammenzug	1x	18x

Tabelle 11: Gegenüberstellung von XP und RUP für den Prozessschritt Lernen

Die Gegenüberstellung zeigt, dass die Stärke von XP in der Schaffung eines lernfreundlichen Arbeitsumfeldes liegt, während die formalisierten Prozesse und die hohe Standardisierung von RUP die Kreativität des Einzelnen tendenziell einengen. Auch das Festhalten des RUP an der Auffassung, dass Fehler früh im Entwicklungsprozess hohe Kosten verursachen, kann ein Klima von Defensivität verbreiten. Damit, dass sich XP von dieser Auffassung befreit, wird den Teammitgliedern Raum zum Experimentieren zugestanden. Missverständnisse und Fehler werden nicht mehr als Problem betrachtet. Auch die Praxis des `Pair Programming` unterstreicht das lernfreundliche Klima: Während der `Driver` sich auf die Implementierung konzentriert, denkt der `Partner` über weitere Konsequenzen nach und bringt andere Ideen ein, so dass das Paar sich ständig weiterentwickeln kann.

Während sich ein neuer Mitarbeiter in einem RUP-Projekt vorwiegend anhand der Dokumentation und mit Hilfe von Schulungen einarbeitet, wird er im XP-Projekt in die Teamprozesse integriert und lernt vor allem als `Partner` in einem Paar. Diese Art zu Lernen wird auch `Shadowing` genannt[110]. Während mit Dokumenten nur explizites Wissen übermittelt werden kann, erlaubt `Shadowing` auch die Vermittlung impliziten Wissens. So lernt der Neuling nicht nur die Projektumgebung kennen, sondern es wird ihm gleichzeitig die Teamkultur nähergebracht, was ihn schnell zu einem vollständigen und verantwortungstragenden Mitglied werden lässt.

Tabelle 12 stellt die Resultate der Gegenüberstellung zusammenfassend dar.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang	RUP	XP
Lernen	1	3	3	2	1x	18x

Tabelle 12: Resultate der Gegenüberstellung für den Prozessschritt Lernen

[107] Interessant ist dabei die Tatsache, dass die Leistung des Paares wesentlich von den Fähigkeiten des Besseren der beiden abhängt (vgl. De Marco et al. 1998, 54).

[108] vgl. Kruchten 2000, 8.

[109] vgl. auch http://www.extremeprogramming.org/stories/pair7.html: Anhang H.

[110] vgl. Pawlowsky et al. 2002, 27.

5.3.4 Einen Beitrag Leisten

Dieser Schritt beschäftigt sich mit den Herausforderungen, die zu bewältigen sind, wenn Wissen, das im Kopf eines einzelnen Mitarbeiters ist, anderen zugänglich gemacht werden soll. Die Teilnehmer der Diagnose haben diesem Schritt in allen drei beurteilten Aspekten hervorragende Bedeutung zugemessen (vgl. Tabelle 13). Besonders wichtig ist er bezüglich Redimensionierungen von Unternehmen, denn nach dem Ausscheiden eines Mitarbeiters bleibt nur zurück, was er an Informationen entweder in Dokumenten erfasst oder anderen Mitarbeitern weitergegeben hat.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang
Einen Beitrag Leisten	2	1	1	1

Tabelle 13: Resultate der Diagnose für den Prozessschritt Einen Beitrag Leisten

Einen Beitrag Leisten kann einerseits bedeuten, sein Wissen zu Papier zu bringen und so einer anderen Person zugänglich zu machen. Dieser Prozess wird Externalisierung genannt[111]. Besonders wirksam erweist sich hier der Einsatz von Metaphern und Analogien:

"Eine Metapher oder eine Analogie repräsentieren eine besondere Art der Wahrnehmung. Sie ermöglicht es Menschen aus verschiedenen Erfahrungswelten, Dinge durch Phantasie und Symbole intuitiv zu begreifen. So kommen sie ohne Analyse und Verallgemeinerungen aus; sie fügen ihr Wissen auf neue Art zusammen und artikulieren etwas, was sie noch nicht in logische Worte fassen können. Dadurch wirkt sich die Metaphorik im Anfangsstadium der Wissensbeschaffung besonders förderlich auf das unmittelbare Engagement für den kreativen Prozess aus."[112]

Ein Mensch kann aber auch ohne Sprache unmittelbar Wissen von anderen erwerben. In diesen Situationen werden Erfahrungen ohne Worte und so ausgetauscht, dass beispielweise gemeinsame mentale Modelle und technische Fertigkeiten entstehen. Diese Art zu lernen findet beispielweise bei der Nachahmung statt. Sie wird durch die Anwendung von Methoden wie Mentorprogramme oder Coaching erleichtert. In der Terminologie des Wissensmanagements heisst sie Sozialisation[113].

In Tabelle 14 sind die Antworten von XP und RUP auf die Herausforderungen des Prozessschrittes Einen Beitrag Leisten aufgeführt.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Motivation		
Barrieren werden abgebaut, die einer Weitergabe im Weg stehen.	Der Prozess regelt die Zusammenarbeit stark. Die formale Organisation erhält hohes Gewicht und stellt sicher, dass keine Barrieren bestehen.	Informelle Netze und Teamgeist stehen im Zentrum. Barrieren werden durch `Pair Programming` und `Collective Code Ownership` abgebaut. Die Praktik des `Customer on site` überbrückt den Graben zwischen Technikern und Fachmitarbeitern.
Wer etwas beiträgt, erhält Chancen und Möglichkeiten zum beruflichen Aufstieg	Die Rollenverteilung hat eine hierarchische Komponente. Somit ist Aufstieg möglich. Es wird vorgeschlagen, die `Best Practices` um die Komponente Teamqualifizierung zu erweitern, um jede Person dort einzusetzen, wo sie sich in der letzten Iteration oder Phase besonders bewiesen hat[114].	XP folgt keinem hierarchischen Gedanken. Die Teams organisieren sich frei. Beruflicher Aufstieg wird nicht thematisiert.
Denjenigen, die nicht beitragen, werden Vergünstigungen vorenthalten.	Hier macht der RUP keine Aussage.	Ist jemand für den Prozess nicht geeignet, so muss er das Team verlassen[115].

[111] vgl. Nonaka et al. 1997, 77.

[112] Nonaka et al. 1997, 23f.

[113] vgl. Nonaka et al. 1997, 75.

[114] vgl. Versteegen 2000, 166.

[115] vgl. Beck 2000, 76.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Der Nutzen für alle Beteiligten wird belohnt.	Der RUP ist der Überzeugung, dass die systematische und schrittweise Erstellung der `Artifacts` in den einzelnen `Workflows` zum Erfolg des Projektes führen wird. Dies ist für ein Projektteam sicher Belohnung genug.	Die einzigen expliziten Produkte sind der Code und die entsprechenden Tests. Die Belohnung ist implizit: Der Code bringt Mehrnutzen für den Kunden, die Tests verleihen Sicherheit darüber, dass der Code in der richtigen Qualität vorliegt.
Erleichterung		
Geben Sie den Beschäftigten Zeit und Raum, um ihre besten Arbeiten als Beitrag zu leisten.	Der Prozess gibt den Teammitgliedern wenig Freiraum. Es werden diejenigen Arbeiten verlangt, die der Prozess von den jeweiligen Rollen erwartet. Trotzdem wird die Fülle der zu erstellenden `Artifacts` genug Gelegenheit bieten, das Beste zu liefern. Dies wird durch den iterativen Ansatz begünstigt.	Qualität ist in XP die unveränderliche Variable. Die Projektmitarbeiter sind aufgefordert, ihr Bestes beizutragen. Dazu dient die Praxis des `Refactoring` und der `Collective Code Ownership`: Wer Verbesserungs- und Vereinfachungspotential erkennt, ist angehalten, dies sofort umzusetzen. Zeit und Raum dazu stehen zur Verfügung.
Schaffen Sie Stellen, die den Beitragsprozess unterstützen.	Der Prozess als Werkzeug unterstützt das Beitragen, indem er festlegt und darstellt, wer wann was abzuliefern hat, wie die Dokumente zu verfassen sind und wo sie abgelegt werden.	Der `Coach` ist der Begleiter des Prozesse und hilft dem Team in allen prozessrelevanten Fragen.
Unterstützen Sie den Transfer impliziten Wissens.	Die visualisierende Modellierung ist ein Kernpunkt sowohl des RUP[116] wie auch des Wissensmanagements. Dies ist sicher einer der bedeutendsten Gewinne, die ein Projekt durch den Einsatz des RUP erhält. Der RUP arbeitet mit einem Visionsdokument. Dies entsteht am Anfang des `Requirements Workflow` und enthält die grundlegenden Bedürfnisse, die mit der Softwarelösung erfüllt werden sollen. Es bringt die Vorstellungen der Kunden zu Papier und will diese den Softwareentwicklern klar machen. Des Weiteren wird während der gesamten Projektdauer ein Glossar gepflegt. Dieses Glossar enthält die Definitionen wichtiger Begriffe fachlicher wie auch technischer Herkunft. Damit will der RUP das gegenseitige Verständnis von Kunden / Endbenutzern und Softwareentwicklern fördern.	In Kapitel 3.4 wurde die Praxis der `Metaphor` erklärt. Diese ist in XP eine Beschreibung des Systems und die Basis für das gemeinsame Verständnis dessen, was das System tun sollte. `Pair Programming` unterstützt den Transfer impliziten Wissens, indem zwei Personen zusammenarbeiten. So findet intensive Sozialisation statt.
Schaffen Sie ein Organisationsnetz (z.B. Mentorprogramme).	Es wird vorgeschlagen, dass Projektleiter als Mentoren arbeiten[117]. Dies bedeutet in der Terminologie des RUP, dass ein erfahrener Projektleiter einen weniger erfahrenen anlernt und ihm die Leitung kontinuierlich überträgt.	Der Big Boss des Projektteams ist auch sein Mentor. `Pair Programming` ist eine Form des `Shadowing`[118].
Vertrauen		
Unterstützen Sie den Austausch auf der Basis von Gegenseitigkeit.	Jede Rolle hat innerhalb eines `Workflows` ihre Verantwortlichkeiten. Dazu gehört die Erstellung von `Artifacts`, die in einem folgenden `Workflow` wiederum als Input dienen.	`Collective Code Ownership` stellt sicher, dass jeder sein Bestes beiträgt und dass jeder von jedem profitieren kann. `Pair Programming`: `Partner` und `Driver` können ihre Rollen tauschen.
Schaffen Sie eindeutige Regeln für die Verwendung intellektuellen Vermögens.	Der Prozess regelt die Verwendung der `Artifacts` durch genaue Zuständigkeiten und Verantwortlichkeiten.	Es gilt die Regel, dass jeder für alles verantwortlich ist und alles verändern darf.
Lassen Sie die Leute ihr Wissen selbst veröffentlichen, um den geistigen Eigentümer zu fördern.	`Artifacts` haben verantwortliche Rollen, die auch für die Veröffentlichung und Versionierung zuständig sind.	`Collective Code Ownership`: der Eigentümer ist das Team.

[116] vgl. Kruchten 2000, 11.

[117] vgl. Versteegen 2000, 175-177.

[118] vgl. Kapitel 5.3.3 Lernen, Seite 42.

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Sorgen Sie dafür, dass sich Vertrauensverhältnisse überschneiden und ergänzen.	Vertrauensverhältnisse basieren auf vertraglichen Grundlagen und auf der Befolgung von formalisierten Abläufen.	Ein XP-Projekt basiert auf Vertrauen. Die `Bill of Rights` bildet die Grundlage des gegenseitigen Vertrauens[119].
Zusammenfassung Einschätzung	10x 26x	25x 7x 4x
Ergebnis Zusammenzug	10x	21x

Tabelle 14: Gegenüberstellung von XP und RUP für den Prozessschritt Einen Beitrag Leisten

Sowohl XP wie auch der RUP bieten vielerlei Antworten auf den Prozessschritt Einen Beitrag Leisten. Ihre Antworten unterscheiden sich aber grundsätzlich: Während der RUP als schwergewichtiger Prozess die Beiträge innerhalb formaler Abläufe ordnet und die Verantwortlichkeiten klar festhält, setzt XP ganz im Sinne der Leichtgewichtigkeit auf face-to-face Kommunikation und geteilte Verantwortlichkeit. In XP ist daher die Zusammensetzung des Teams für den Erfolg des Projektes zentral: Das Team muss in dem Sinne eingeschworen sein, dass die Mitglieder untereinander eine enge und vertrauensvolle Beziehung haben[120]. Dies bietet einem XP-Projekt die Chance, dass das Ganze grösser als die Summe seiner Teile ist: Die Ergebnisse, die dieses Team erreichen kann, sind besser als die Ergebnisse, die das gleiche Team produzieren kann, wenn jeder für sich alleine arbeitet. In einem Projekt, das mit RUP arbeitet, ist dies weniger wichtig: Die Zusammenarbeit funktioniert auch ohne derart enge Beziehungen, denn sie ist in den `Workflows` geregelt. Allerdings fehlt dadurch auch das Potential zu echtem Team-Lernen, das von offener, forschender und vertrauensvoller Kommunikation abhängt[121].

Bezüglich Redimensionierungen von Unternehmen bietet der RUP den Vorteil, dass die Informationen in Form vielfältiger `Artifacts` vorliegen. Der Prozess ist damit weniger von den einzelnen Personen abhängig als bei XP, wo ein viel grösserer Teil des Wissens nur in den Köpfen der Teammitglieder verankert ist. Der Code und die entsprechenden Tests bilden die einzigen externalisierten Informationen. Da Redimensionierungen das eingeschworene Team verändern, muss der Teambildungsprozess wieder anlaufen, was die Produktivität kurzfristig reduziert.

Die Zusammenfassung der Resultate der Gegenüberstellung in Tabelle 15 zeigt, dass beide Prozesse gut abschneiden – wenn auch mit sehr unterschiedlichen Methoden.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang	RUP	XP
Einen Beitrag Leisten	2	1	1	1	10x	21x

Tabelle 15: Resultate der Gegenüberstellung für den Prozessschritt Einen Beitrag Leisten

[119] vgl. Beck 2000, 147f, Jeffries et al. 2001, 7, Kapitel 3.5.

[120] zum Begriff des eingeschworenen Teams vgl. De Marco 1998, 143f.

[121] vgl. Willke 2001, 50f und Senge 2001, 284-289.

5.4 Die strategischen Schritte

5.4.1 Beurteilen

Die Diagnose hat gezeigt, dass dem Prozessschritt Beurteilen im Rahmen der Softwareentwicklung nur geringe Bedeutung zugemessen wird (vgl. Tabelle 16).

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang
Beurteilen	7	6	6	6

Tabelle 16: Resultate der Diagnose für den Prozessschritt Beurteilen

Tabelle 17 zeigt die Imperative und Herausforderungen dieses Prozessschrittes.

Imperative und Herausforderungen
Perspektive
Identifizieren Sie neue Formen unternehmerischen Kapitals
Entwickeln Sie Konzepte für die neuen Managementaufgaben
Integration
Visualisieren Sie die zugrunde liegenden Strukturen, welche die Wissensmanagement-Praktiken steuern
Experimentieren Sie mit Massstäben und Beurteilungsansätzen, um die strategischen Ergebnisse zu bewerten
Kommunizieren Sie mit den Leuten, die ein berechtigtes Interessse am Unternehmen haben

Tabelle 17: Die Imperative und Herausforderungen des Prozessschrittes Beurteilen

Weder XP noch RUP bieten bezüglich der Bewältigung dieser Herausforderungen Hinweise, weshalb die Tabelle lediglich eine Spalte hat. Beide bieten aber die Grundlage dafür, dass die in der Software enthaltenen Komponenten als unternehmerisches Kapital gemessen werden können. Unternehmerisches intellektuelles Kapital lässt sich den beiden Kategorien Humankapital und strukturelles Kapital zuteilen[122]:

1. Humankapital = an Personen gebundenes Wissen; Kompetenzen, Denkweisen, Problemlösefähigkeit von Teams und Einzelpersonen.
2. Strukturelles Kapital = organisatorisch verankertes Wissen; Informationen, die elektronisch verfügbar sind und Wissen, das in Prozessen umgesetzt ist.

Da der RUP bedeutend mehr dinglichen Output herstellt, ist dort das Strukturkapital höher zu bewerten als dasjenige eines Projektes, das mit XP arbeitet. In letzterem ist dafür das Humankapital höher zu bewerten. Beide Prozesse haben aber zum Ziel, qualitativ hochstehendes strukturelles Kapital zu bilden, indem Geschäftsprozesse in Software abgebildet werden.

Abschliessend ist festzuhalten, dass Beurteilen eine Aufgabe des Managementumfeldes des Softwareprozesses ist, dass dieser aber Daten für die Beurteilung liefern kann. Anhand der Diagnose wurde festgestellt, dass ein derartiges Messen als hinderlich empfunden werden kann.

5.4.2 Aufbauen und Pflegen

Dem Prozessschritt Aufbauen und Pflegen kommt den Resultaten der Diagnose zufolge zentrale Bedeutung bei Entlassungen und Einstellungen zu (vgl. Tabelle 18).

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang
Aufbauen und Pflegen	4	2	2	3

Tabelle 18: Resultate der Diagnose für den Prozessschritt Aufbauen und Pflegen

Tabelle 19 bietet eine Übersicht über die Antworten von RUP und XP auf die Herausforderungen dieses Prozessschrittes.

[122] vgl. Probst et al. 1998, 26.

Imperative und Herausforderungen	Antwort von RUP	Antwort von XP
Richtung		
Die Informationstechnik wird den Menschen untergeordnet.	Der Werkzeugunterstützung wird im Verhältnis zur menschlichen Komponente zu viel Gewicht beigemessen. Allerdings legt der RUP Gewicht auf die Anforderungen des Menschen, was im Workflow des Requirements Management deutlich wird. 	Der Mensch mit seiner Individualität und Kreativität steht im Zentrum.
Positionen werden so strukturiert, dass sich die Aufmerksamkeit des Unternehmens auf geistige Vermögenswerte richtet.	Der Einsatz eines so teuren und umfassenden Prozesses wie RUP erteilt dem Softwareprozess als aus geistigem Eigentum Mehrwert schaffenden Prozess Gewicht. 	Die betonte Leichtgewichtigkeit von XP vermag es u.U. nicht, dem Softwareprozess innerhalb der Unternehmung genug Gewicht zukommen zu lassen, da er kaum mit Kosten und kaum mit objektiven Produkten ausser dem lauffähigen Programm verbunden ist.
Verbindung		
Predigen Sie die Zusammenarbeit von verschiedenen Abteilungen im Unternehmen.	Die Notwendigkeit der guten Zusammenarbeit zwischen Technikern und Fachabteilung wird betont und ist in der 4+1-View verinnerlicht. Des Weiteren sind vertragliche Abmachungen und formale Arbeitsabläufe institutionalisiert. 	Die Zusammenarbeit basiert auf gegenseitigem Verständnis, Vertrauen und dem Wahrnehmen von Verantwortlichkeiten. Diese sind in Übereinstimmung mit der `Bill of Rights` akzeptiert.
Finden Sie neue Wege der Partnerschaft mit anderen Firmen.	Weder XP noch RUP machen hier konkrete Aussagen. Durch die starke Web-Präsenz der Fangemeinden und der Autoren beider Prozesse ermöglichen aber beide das Finden von Partnerschaften und den unternehmensübergreifenden Austausch von Wissen. 	
Halten Sie die richtigen Leute im Unternehmen.	Im RUP werden die verschiedenen Phasen mit unterschiedlicher Anzahl von Personen besetzt[123]. Dies verhindert die Bildung eines eingeschworenen Teams und ist diesem Ziel nicht förderlich. 	XP stellt klare Anforderungen an die Fähigkeiten der Mitglieder des Teams. Zentral ist dabei die Kommunikationsfähigkeit. Personen, welche diese Fähigkeiten nicht mitbringen, müssen das Team verlassen. Das Tempo des Teams wird vom Team selbst bestimmt. Es ist nicht erlaubt, zu müde zu werden und sich die Energie von morgen zu leihen, um heute etwas mehr zu erledigen[124]. Dies verhindert Burnout. Die Betonung darauf, dass Arbeit Spass machen sollte, ist diesem Ziel sicher ebenfalls zuträglich.
Erkenntnis		
Nutzen Sie Wissen, um die Beziehung zum Kunden zu festigen.	Grosses fachliches Know-how wird durch das Erarbeiten der Business-Anforderungen gewonnen. Dadurch wird Wissen geschaffen, das die Beziehung zum Kunden fördert. 	Die enge Zusammenarbeit zwischen Technikern und Fachabteilungen lässt eine starke emotionale Bande zwischen den Beteiligten entstehen.
Nehmen Sie das Unternehmen als Ganzes auseinander, um die Bestandteile in neuem Licht zu sehen.	Der RUP integriert einen stark formalisierten und mächtigen `Workflow` für die Aufnahme und Bewirtschaftung von Anforderungen. So bietet er bezüglich dieser Herausforderung und den betrachteten Businessprozessen enormes Potential. Dadurch, dass sich Rollen explizit nur um das Modellieren von Geschäftsprozessen kümmern, kann viel Energie zur Bewältigung dieser Herausforderung eingesetzt werden. 	XP als leichtgewichtiger Entwicklungsprozess arbeitet bei der Anforderungsanalyse mit `Story Cards`. Diese beschreiben aus Kundensicht übersichtsartig bestimmte Anwendungsfälle, die das System zulassen soll. Eine ganzheitliche, den ganzen Geschäftsprozess hinterfragende Sichtweise wird in XP nicht gefördert.

[123] vgl. Kruchten 2000, 128.

[124] vgl. Cockburn et al. 2001.

Gegenseitigkeit		
Demonstrieren Sie, dass die Schaffung von Nutzen sich für alle Beteiligten lohnt.	Der Prozess verspricht, dass seine Befolgung zu Projekterfolg führt.	Die Praktiken des Testens und des `Refactoring` unterstützt diese Herausforderung. Jeder Beitrag nützt dem ganzen Team, indem neue Funktionalität in der richtigen Qualität entsteht.
Schaffen Sie ein Umfeld, in dem der ganze Mensch täglich zur Arbeit antreten kann.	Der RUP ist vor allem auf erfolgreiches Projektmanagement ausgerichtet und befasst sich kaum mit den weichen Komponenten des Arbeitsumfeldes. Die Kreativität wird durch die hohe Standardisierung eingeschränkt.	Die Arbeit mit XP soll Spass machen, die Individualität und Kreativität jedes Einzelnen steht im Zentrum.
Zusammenfassung Einschätzung	14x 10x 3x	16x 9x 2x
Ergebnis Zusammenzug	11x	14x

Tabelle 19: Gegenüberstellung von XP und RUP für den Prozessschritt Aufbauen und Pflegen

Wiederum zeigt sich die grundlegende Andersartigkeit der Prozesse: Während der RUP vorwiegend elektronische Medien verwendet, wird in XP die menschliche Komponente ins Zentrum gerückt. Als schwergewichtiger Prozess könnte der RUP dem Wissensmanagement helfen, im organisatorischen Umfeld mehr Bedeutung zu erlangen, da er bedeutende Investitionen und Aufmerksamkeit erfordert. XP, dessen Einsatz kaum Investitionen benötigt und vor allem auf die weichen Faktoren angewiesen ist, erlangt in konventionellen Managementkonzepten demgegenüber weniger Beachtung.

Was das Halten von Personen im Unternehmen betrifft, ist der RUP deutlich schwächer als XP. Da im RUP die Zusammensetzung des Teams im Ablauf der Phasen verändert wird und die Teammitglieder dedizierte Rollen wahrnehmen, werden die Personen weder durch starke Teambeziehungen emotionell gebunden, noch erleben sie ihre Aufgaben als abwechslungsreich. Zudem schränkt die starke Formalisierung und Standardisierung des Prozesses die Kreativität ein.

In Tabelle 20 sind die Resultate der Gegenüberstellung zusammengefasst.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang	RUP	XP
Aufbauen und Pflegen	4	2	2	3	11x	14x

Tabelle 20: Resultate der Gegenüberstellung für den Prozessschritt Aufbauen und Pflegen

5.4.3 Aussondern

Wie Tabelle 21 zeigt, messen die Teilnehmer der Diagnose dem Prozesschritt Aussondern wenig Bedeutung zu.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang
Aussondern	5	7	7	6

Tabelle 21: Resultate der Diagnose für den Prozessschritt Aussondern

Tabelle 22 zeigt, dass sowohl der RUP als auch XP einige wertvolle Antworten auf die Herausforderungen dieses Porzesschrittes bereit halten. Beide geben Hinweise auf die Fähigkeiten, welche die Teammitglieder für die Wahrnehmung der jeweiligen Rollen mitbringen sollten. Grundsätzlich gehen die Modelle beider Prozesse davon aus, dass die notwendigen Fähigkeiten im Projektteam selbst entwickelt und nicht von aussen hereingeholt werden sollten. Dies ist mit den Ergebnissen der Diagnose vereinbar: Die Aussage "Wir holen

Fähigkeiten und Fachkenntnisse von aussen herein, wenn sie nicht zu unseren Kernkompetenzen gehören" wurde häufig als "wäre schädlich" markiert[125].

Imperative und Herausforderungen	**Antwort von RUP**	**Antwort von XP**
Verzicht		
Man unterscheidet zwischen Arten von Wissen, die ausgenutzt werden können, und solchen, die nur von begrenztem Wert sind.	Es werden nur Aussagen dazu gemacht, was man braucht, nicht dazu, was man nicht braucht oder schädlich ist.	Es werden Rollen definiert. Es wird ausgesprochen, welche Fähigkeiten nicht gut sind, und dass man solche loswerden sollte.
Es werden Alternativen zum direkten Erwerb von Wissen gefunden, so dass man damit experimentieren kann.	Outsourcing wird im RUP dann thematisiert, wenn es darum geht, mit Terminkonflikten umzugehen[126]. Hier wird empfohlen, zu entscheiden, ob gewisse Tätigkeiten ausgelagert werden können.	Externe Mitarbeiter sind grundsätzlich nur für die Schulung von solchen Fertigkeiten empfohlen, die zur Zeit im Projektteam noch nicht vorhanden sind. Externe Mitarbeiter sollen ihr Wissen nicht einsetzen, ohne dass sie es gleichzeitig an interne Projektmitarbeiter weitergeben.[127] `Outsourcing` wird mit wenig Begeisterung aufgenommen. Stattdessen `Insourcing` vorgeschlagen[128]: Teammitglieder aus den Reihen des Auftragnehmers werden kontinuierlich mit solchen aus den Reihen des Kunden ersetzt. So kann der Know-how-Transfer zwischen Auftragnehmer und Kunden stattfinden und der Kunde wird sich nicht mit einem System wiederfinden, von dem er nicht weiss, wie es arbeitet. Dieses Vorgehen könnte dieser Herausforderung sowie den Ansprüchen der Beurteilenden gerecht werden.
Umwandlung		
Echte Belastungen für die Ressourcen werden erkannt und abgeschafft.	Weder im RUP noch in XP konnten zu dieser Herausforderung Aussagen gefunden werden.	
Man sollte nicht das Kind mit dem Bade ausschütten.	Es wird nichts weggeworfen.	Durch die Philosophie "Design for today" besteht die Gefahr, dass etwas weggeworfen wird, was man später noch gebrauchen könnte. Durch das Führen des Respositories besteht aber die Möglichkeit, alte Stände jederzeit wieder zurückzuholen.
Zusammenfassung Einschätzung	2x 8x 2x	6x 4x 2x
Ergebnis Zusammenzug		4x

Tabelle 22: Gegenüberstellung von XP und RUP für den Prozessschritt Aussondern

Die Resultate der Gegenüberstellung sind in Tabelle 23 zusammengezogen.

Prozessschritt	Rang SW-Prozess	Rang Entlassung	Rang Einstellung	Total Rang	RUP	XP
Aussondern	5	7	7	6		4x

Tabelle 23: Resultate der Gegenüberstellung für den Prozessschritt Aussondern

5.5 Fazit

Nachdem jeder Prozessschritt untersucht wurde, werden nun die beiden Prozesse aus der Vogelperspektive betrachtet. So wird es möglich sein, einige Empfehlungen abzugeben, deren Umsetzung dabei hilft, Wissensmanagement im Softwareprozess umzusetzen. Es soll erreicht

[125] vgl. dazu Tabelle 3, Seite 18 und Anhang C.
[126] vgl. RUP 2000 eCoach, Guidelines zur Business Vision.
[127] vgl. Beck 2000, 146f.
[128] vgl. Beck 2000. 160-162.

werden, dass der Softwareprozess die Anliegen des Wissensmanagements unterstützt, dass neuen Mitarbeitern den Weg zur Produktivität leicht fällt und dass das verbleibende Team nach Redimensionierungen unter möglichst wenig Know-how-Verlust leidet.

Bei der Beurteilung jedes Prozessschrittes wurde immer wieder klar, dass bedingt durch die grundsätzliche Andersartigkeit der beiden Entwicklungsprozesse auch jede Herausforderung des Wissensmanagements auf unterschiedliche Art bewältigt wird. Es hat sich gezeigt, dass die Art des einen Vorgehensmodells im täglichen Entwicklungsprozess förderlich erscheint, während die andere bedeutende Vorteile bietet, wenn es um Einarbeitungen und Entlassungen geht. Wie Tabelle 24 zeigt, hat XP gegenüber RUP in allen Wissensmanagement-Prozessschritten Vorteile. Es hat sich erwiesen, dass sich XP und seine starke Ausrichtung auf Kommunikation für Wissensentwicklung, -austausch und -beitrag nützlicher erweist als der RUP, der sich an Werkzeugen und Formalisierung hält.

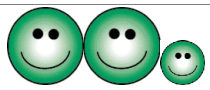
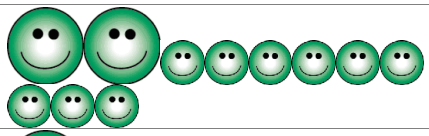
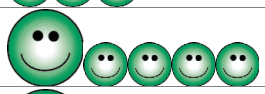
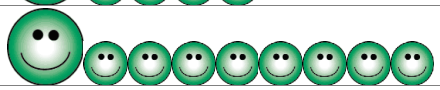
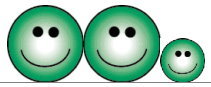
Prozessschritt	SW-Prozess	Ent-lassung	Ein-stellung	Total Rang	RUP	XP
Beschaffen	6	5	4	5		
Nutzen	3	4	5	4		
Lernen	1	3	3	2		
Einen Beitrag Leisten	2	1	1	1		
Beurteilen	7	6	6	6	-	-
Aufbauen und Pflegen	4	2	2	3		
Aussondern	5	7	7	6		

Tabelle 24: Zusammenfassung der Gegenüberstellung: Ein grosses Symbol steht für 10 kleine.

(Beginn Modifikation durch Hans-Peter Korn am 27. Jan. 2004)

Abbildung 18 stellt dar, wie gut die verlangte Abdeckung der WM-Schritte von RUP bzw XP erfüllt wird. 100% bedeutet, dass das, was für die Fälle „Softwareentwicklung“, „Entlassung“ oder „Neueinstellung“ verlangt wird (siehe Abbildung 8: Medianwerte der veralngten Abdeckung der WM-Bereiche) vollständig erfüllt wird.

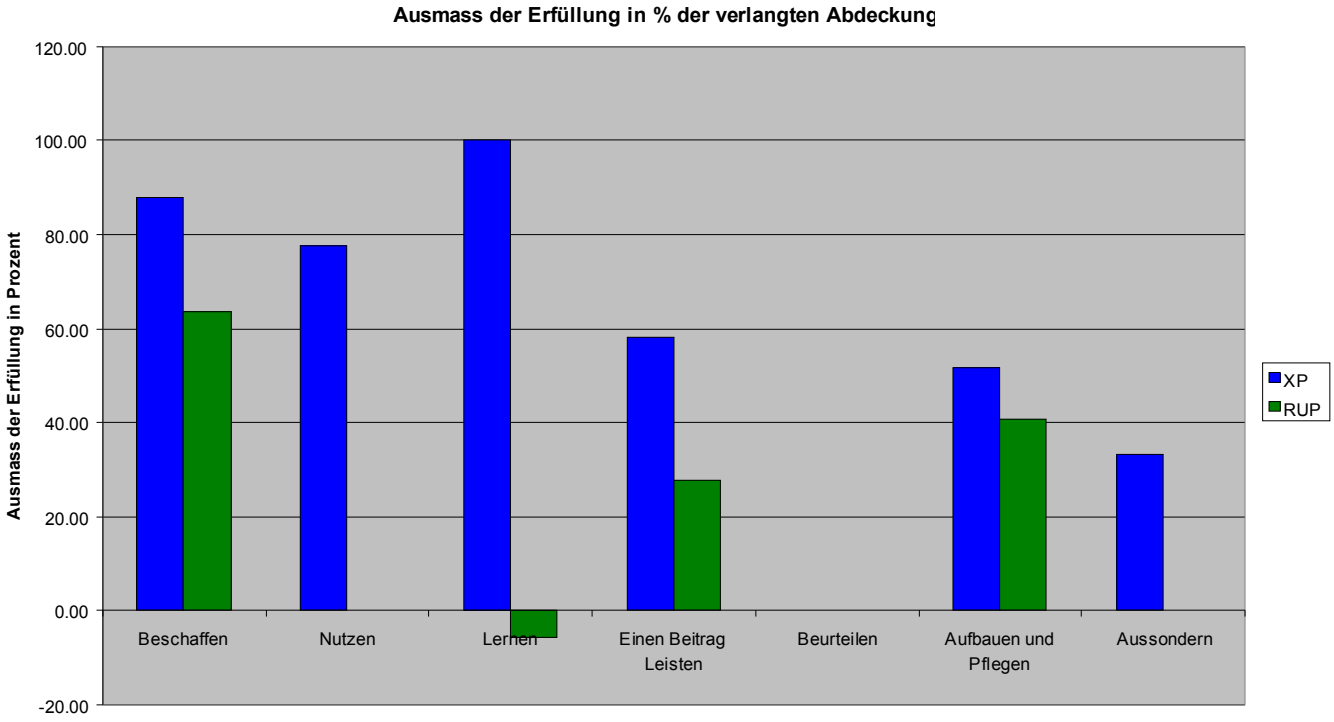


Abbildung 18: Balkendiagramm der Erfüllung der verlangten Abdeckung der einzelnen WM-Bereiche durch XP und RUP

(Ende dieser Modifikation)

- Lernen – Ins Auge fällt sofort der unterschiedliche Abdeckungsgrad des Bereiches "Lernen". Die Beurteilung ergab, dass RUP das Lernen nicht fördert, sondern ihm sogar hinderlich ist. Dies ist vor allem darin begründet, dass er stark formalisiert ist und daher weder Eigenverantwortung noch individuelle Denkanstrengungen fördert. XP hingegen schliesst Praktiken und Prinzipien mit ein, welche dem Lernen enorm zuträglich sind: Pair `Programming`, `Standup Meeting`, `Planning Game` und `Refactoring`.

- Nutzen – Frappant ist ebenfalls der Balken des Bereiches "Nutzen". Der RUP wurde hier völlig neutral eingestuft – er scheint keine Hilfestellungen zu geben, die diesem Wissensmanagementbereich zugute kämen. XP bietet auch hier den besseren Rahmen.

- Beschaffen – Den Prozesschritt "Beschaffen" unterstützen beide Modelle auf ihre Weise. Während der RUP auf Schriftgut setzt, sind die Mitarbeiter in XP dazu aufgefordert, miteinander zu sprechen. Im RUP werden die zu erstellenden Dokumente in Workflows definiert und Eigentümern zugeteilt. In XP dagegen ist der Mitarbeiter darauf angewiesen, die Wissensträger zu kennen.

- Einen Beitrag Leisten – Den Wissensmanagementbereich "Einen Beitrag Leisten" meistern beide Modelle in unbefriedigender Weise, denn keiner gibt Empfehlungen zu Regeln bezüglich der Verwendung der Beiträge ab. XP hat zudem die Schwäche, dass es nicht möglich ist, die Beiträge bestimmten Personen zuzuordnen. *Dennoch unterstützt XP diesen WM-Aspekt deutlich besser als RUP (Anm. H.-P. Korn, 27.01.04)*: Der RUP baut stark auf Formalismen und wirkt daher starr. Er wird die Mitarbeiter nicht motivieren, ungewöhnliche Beiträge zu leisten.

- Aufbauen und Pflegen – Beide Prozessmodelle unterstützen das Wissensmanagement ähnlich gut, wenn auch, bedingt durch die schon mehrmals angesprochene Differenzen in der ihnen zugrundeliegenden Philosophie, auf völlig verschiedene Art und Weise.

- Aussondern – Die Beurteilung von RUP wurde in diesem Bereich als völlig neutral eingestuft. Er verhindert Aktivitäten, noch bietet er dafür einen Rahmen. XP hingegen enthält Praktiken und Prinzipien, welche dem Aussondern zuträglich sind: Im `Pair Programming` werden alte Denkmuster reflektiert, es wird Wissen entwickelt und altes Wissen fallengelassen. Bezüglich der Projektmitarbeiter nehmen die Vertreter von XP eine klare Haltung ein: Wer das Team stört oder die Arbeitsweise nicht akzeptieren kann, der muss es verlassen.

Die Abbildungen Abbildung 19 bis Abbildung 21 illustrieren das Verhältnis zwischen verlangter Abdeckung der Bereiche des Wissensmanagements und der ermittelten Abdeckung.

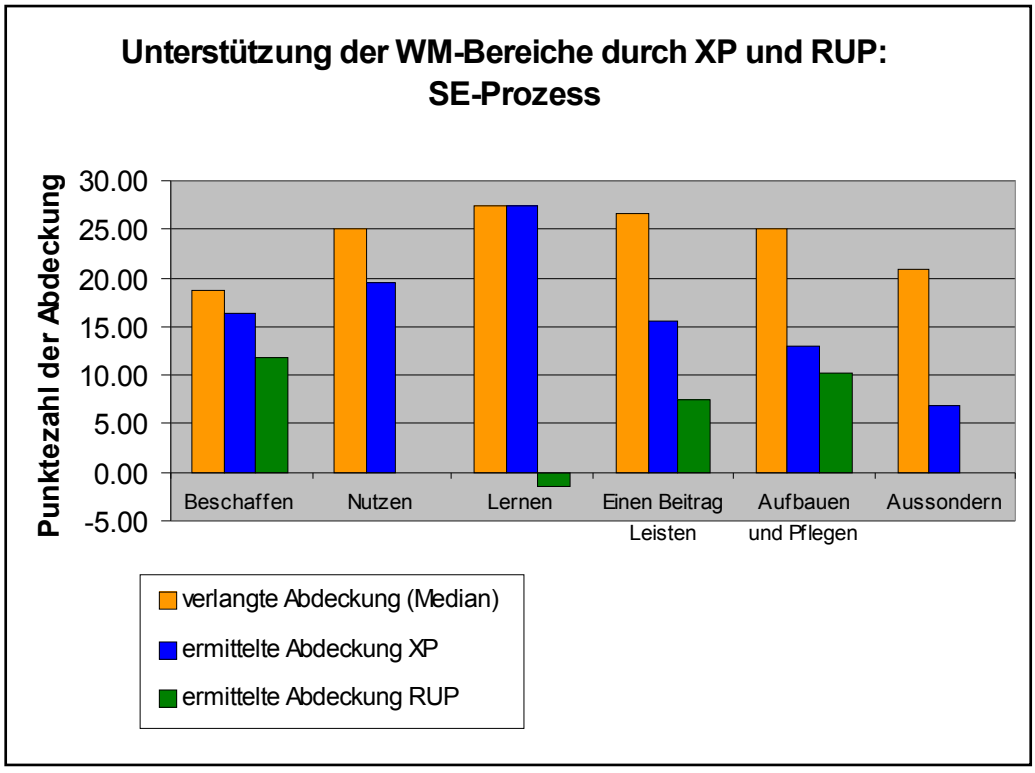


Abbildung 19: Softwareentwicklungsprozess: verlangte und ermittelte Abdeckung der WM-Bereiche

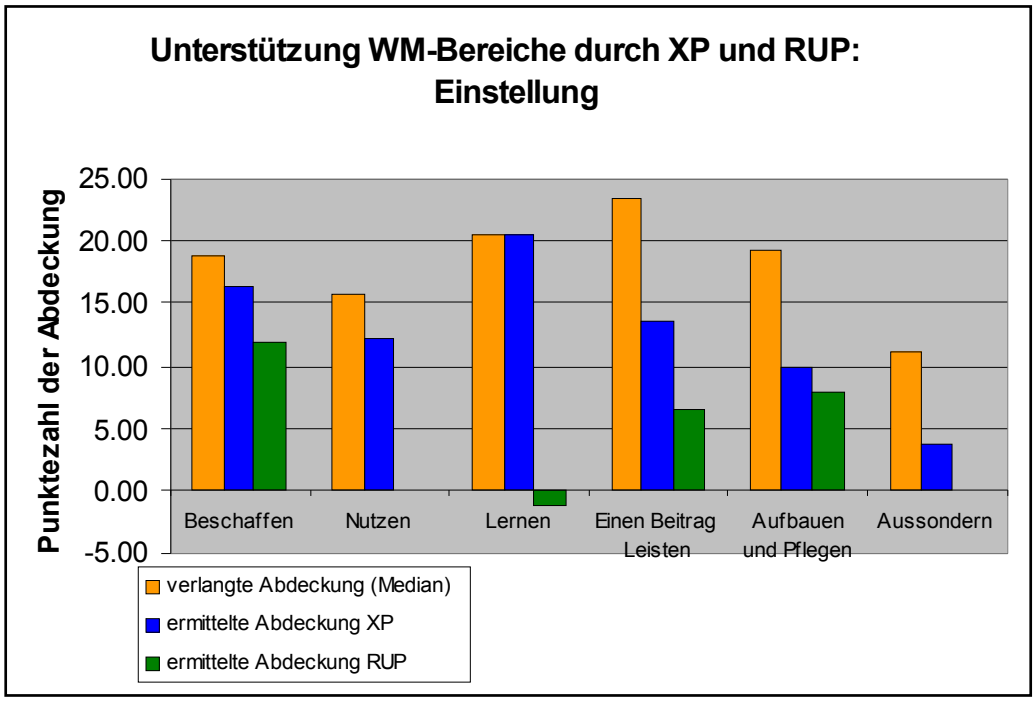


Abbildung 20: Einstellungen: verlangte und ermittelte Abdeckung der WM-Bereiche

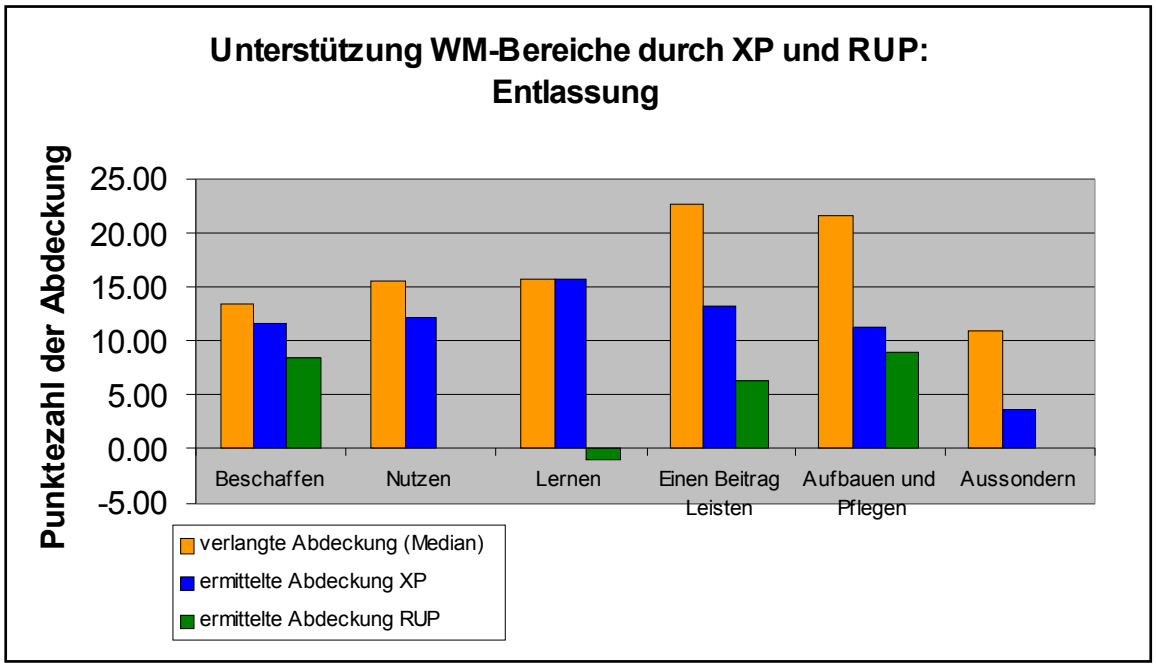


Abbildung 21: Entlassungen: verlangte und ermittelte Abdeckung der WM-Bereiche

Sowohl im täglichen Geschäft der Softwareentwicklung wie auch in den Ausnahmesituationen, wenn Personen entlassen oder eingestellt werden, wurde "Einen Beitrag Leisten", "Aufbauen und Pflegen" und "Aussondern" eine Bedeutung zugeordnet, welche beide Prozessmodelle nicht bewältigen können. XP schneidet aber dennoch etwas besser ab – der RUP bietet fast keine Unterstützung.

Immer wieder wurde in der Beurteilungsphase deutlich, wie unterschiedlich die beiden Prozessmodelle sind. Sowohl die Dokumentenorientierung des einen wie auch die Fokussierung auf Kommunikation und Interaktion haben Vorteile. Auch wenn der RUP aus der Vogelschau schlechter abschneidet, sollten nicht all seine Ansätze in den Wind geworfen werden, denn so einleuchtend uns die folgende Aussage erscheint: "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation"[129], so gefährlich erweist sich das strenge Vorgehen nach diesem Prinzip, wenn es um Entlassungen geht. Wenn eine Person das Team verlässt, ist alles Wissen, das nur im Kopf dieses Einzelnen verankert ist, mit seinem Weggang für den Rest des Teams verloren. Daraus folgt die Empfehlung, einen gewissen Grundschatz an Dokumentation zu erstellen. Bei der Erstellung dieses Grundstockes sollte nach dem Grundprinzip des RUP wo immer möglich mit visualisierenden Modellen gearbeitet werden, welche nach der UML-Notation als Standard der objektorientierten Welt verfasst werden. Diese Dokumente müssen selbstverständlich gewartet werden – je nach Arbeitsweise des Teams mag es sich empfehlen, dafür einen Verantwortlichen zu bestimmen oder das Dokument der Verantwortlichkeit aller zu überlassen.

Wie schon diese erste Empfehlung zeigt, ist es aus Sicht des Wissensmanagers sinnvoll, Elemente der beiden Prozesse zu integrieren[130]. Während es sinnvoll ist, aus dem RUP eine Auswahl von `Artifacts` zu übernehmen, sollten von XP einige Praktiken übernommen werden, die dem Wissensaustausch und dem Lernen besonders förderlich sind:

- `Pair Programming`: Arbeiten zwei Personen gemeinsam, so erstellen sie ein Produkt besserer Qualität als wenn jeder alleine arbeitet. Nicht jede Arbeit scheint aber geeignet für die Arbeit zu zweit. Die Praktik sollte daher differenziert angewendet werden: Erfahrungen haben gezeigt, dass sie für Bugfixing oder Wartung wenig Mehrwert schafft[131].
    - `Design Simplicity` und `Refactoring`: Wie bei `Pair Programming` handelt es sich auch hier um eine Praktik, die während der Entwicklungstätigkeit angewendet werden kann und die deutlich zur Qualitätssteigerung beiträgt.
    - `Testing`: Sowohl XP wie auch RUP legen grossen Wert auf die Qualität der Software. Der `Test Workflow` des RUP erstreckt sich über alle Phasen des Projektes. Die hervorragende Betonung von `test first` ist der einzige signifikante Unterschied von XP gegenüber dem RUP. Auch diese Praktik bezieht sich auf die Entwicklungstätigkeit und trägt zur Steigerung der Qualität, Sicherheit der Entwickler und zur Ausbreitung von Know-how bei.
    - `Customer on site`: Befindet sich ein Kunde in demselben Raum wie das Entwicklerteam, so fördert dies nicht nur die persönlichen Beziehungen, sondern auch den Wissensaustausch. Die Entwickler erlangen ein gründlicheres Verständnis für die Bedürfnisse der Kunden, als wenn sie den Kunden nur in gelegentlichen Meetings treffen. Es kann schwierig sein, diese Praktik umzusetzen, da der Kunde das Entwicklungsprojekt oft aus der Perspektive des Auftraggebers betrachtet und sich nicht als Teammitglied sieht. Hier liegt es am

[129] Martin 2001, 12.

[130] Für RUP existiert auch bereits ein käufliches XP-Plugin: http://www.rational.com/demos/viewlets/rup/xp_plugin/XP_PlugIn_Preview_viewlet.html (am 23. Juni 2003)

[131] vgl. Elssamadisy, Anhang I, Seite 4.

Projektmanager, den Kunden in das Team zu integrieren und dafür zu sorgen, dass er die Bedeutung seines Wissens für die Entwickler erkennt.

Alle diese Praktiken unterstützen die Entwicklungstätigkeit, für die der RUP neben den Werkzeugen wenig Unterstützung bietet.

Dass beide Prozesse die Herausforderungen des Wissensmanagements auf ihre Art meistern können, vermag nicht zu überraschen, wenn wir uns an ihre jeweiligen Zielsetzungen erinnern: Beide deklarieren das Vermeiden von Risiken und die Verbesserung der Qualität des erstellten Produktes als Grund dafür, dass Software überhaupt gemäss einem Vorgehensmodell entwickelt werden soll. Beide gehen daher auf ihre Weise auf dieselbe Grundproblematik der Softwareentwicklung ein: Zielkonflikte und fachliche Verständnisschwierigkeiten zwischen Technikern und Fachspezialisten. Diesen Grundproblemen kann besser begegnet werden, wenn wir die Brille des Wissensmanagers aufsetzen. Doch impliziert diese Nähe von Softwareentwicklung und Wissensmanagement auch, dass die Ursachen für das Versagen eines Wissensmanagementprozesses auch die Ursachen für das Versagen eines Entwicklungsprojektes sein können[132]:

1. Man verlässt sich zu sehr auf einfach zu findende und darzustellende Information und verliert sich in deren Quantität statt Qualität.
2. Man kümmert sich zu sehr um Werkzeuge.
3. Man setzt für das Managen eine Rolle ein, die zu weit vom Tagesgeschäft entfernt ist.

Ein RUP-Projekt ist all diesen Gefahren ungleich mehr ausgesetzt als ein XP-Projekt. Es ist daher unumgänglich, dass der RUP entsprechend den Bedürfnissen des Projektteams angepasst wird. Die Steuerung des Prozesses ist daher in RUP als genauso zentral anzusehen wie die Projektsteuerung selbst.

Abschliessend lässt sich festhalten, dass wie im Wissensmanagement auch im Softwareprozess ein geeigneter Mix zwischen Personalisierung und Dokumentation gefunden werden muss.

[132] vgl. Von Krogh et al. 2000, 26-29.

6 Zusammenfassung

Die Absicht dieser Arbeit war es, die beiden Prozessmodelle RUP und XP hinsichtlich ihrer Fähigkeit zu untersuchen, Praktiken des Wissensmanagements zu integrieren. Dabei standen drei Situationen im Entwicklungsprojekt zur Diskussion: Der Entwicklungsprozess wurde unter der Annahme betrachtet, dass erstens das Team konstant bleibt, dass zweitens Personen das Team verlassen und dass drittens neue Personen die Arbeit im Team aufnehmen. Um festzustellen, welche Aspekte des Wissensmanagements in welcher Situation besonders wichtig sind, wurde eine Diagnose mit 10 Personen durchgeführt. Diese Personen beurteilten eine Sammlung von Aussagen hinsichtlich ihrer Relevanz für den Softwareentwicklungsprozess. Die Aussagen waren in 7 Abschnitte gegliedert, wobei sich drei der Abschnitte mit dem strategischen Teil des Wissensmanagements befassten: Beurteilen, Aufbauen und Pflegen und Aussondern. Die restlichen vier betrafen den taktischen Teil: Lernen, Nutzen, Einen Beitrag Leisten und Beschaffen.

Die Diagnose hat gezeigt, dass Einen Beitrag Leisten von zentralster Bedeutung ist. Dies bedeutet, dass das Prozessmodell die Mitarbeitenden einerseits in ihrer Freude unterstützen soll, ihr Wissen weiterzugeben. Andererseits sollte es Systeme und Strukturen integrieren, die das Beitragen erleichtern und Respekt und Verständnis für den Wert der Beiträge fördern.

An zweiter Stelle folgt Lernen. Das gewählte Prozessmodell sollte den freien Fluss von Ideen unterstützen und Information als offene Ressource betrachten. Damit auf Ideen selbstbewusst reagiert wird, müssen Hemmungen abgebaut werden. Ausserdem sollte Lernen im täglichen Handeln gefördert werden, zum Beispiel durch `Learning by Doing`. Lernen wird in einem solchen Umfeld zur Gewohnheit, denn der Mitarbeitende ist in der täglichen Arbeit zum Mitdenken aufgefordert und darf Fehler begehen, um aus ihnen lernen zu können.

Ein weiterer wichtiger Aspekt ist Aufbauen und Pflegen. Hier sollen die Ressourcen durch den Einsatz des Prozessmodells so kanalisiert werden können, dass Wissen geschaffen und vermehrt wird. Dazu müssen Beziehungen aufgebaut werden, welche dem Ziel der Wissensentwicklung zuträglich sind. Des Weiteren ist es wichtig zu erkennen, welcher Nutzen aus bestehendem Wissen herausgefiltert werden kann. Dies setzt voraus, dass das Prozessmodell eine Kultur unterstützt, welche den Bund zwischen Mitarbeitern und Unternehmen festigt, da der Mitarbeiter nur dann motiviert sein wird, sein Denken zugunsten des Unternehmens einzusetzen.

Sowohl RUP wie auch XP bieten Antworten auf diese Herausforderungen, jedoch auf sehr verschiedene Art und Weise. Dies hat seinen Ursprung in den folgenden beiden grundlegenden Unterschieden zwischen dem RUP und XP:

1. Die erste augenfällige Differenz zwischen den beiden Prozessmodellen ist, dass der RUP sich vor allem mit den Aufgaben des Projektmanagements befasst, während XP aus diesem Bereich nur die Planung genauer beschreibt. XP legt dafür sein Hauptaugenmerk auf die Entwicklungstätigkeit, wo der RUP wiederum nur sehr spärliche Hilfestellungen leistet.
2. Der zweite grosse Graben tut sich bezüglich des Umfangs der Prozessmodelle auf: Während RUP ganz im Sinne der traditionellen, schwergewichtigen Prozesse Wert auf das Befolgen von genau definierten Abläufen legt, in denen zu definierten Zeitpunkten spezifizierte Dokumente entstehen, folgt XP dem Grundsatz, dass der Code und seine Tests das System ausreichend dokumentieren und lediglich acht Praktiken befolgt werden müssen. Dabei setzt XP vollständig auf die kommunikative Zusammenarbeit im Projektteam.

Beide Ansätze haben ihre Stärken. Die Betonung auf formalisierte Teilprozesse unterstützt vor allem den Schritt Aufbauen und Pflegen. Damit garantiert der RUP, dass Personen zwischen den Projekten ausgetauscht werden können, weil sie sich in der neuen Umgebung

schnell zurechtfinden. Der auf Kommunikation setzende Ansatz von XP ist hingegen vor allem im Gebiet des Lernens und Nutzens stark, indem er die Chance bietet, eingeschworene Teams entstehen zu lassen. Mitglieder solcher zusammengeschweisster Arbeitsgruppen sind tendenziell innovativer und produktiver, da sie sich weitgehend selbst organisieren und ihre individuelle Kreativität stärker einzubringen vermögen. Den Aspekt von Einen Beitrag Leisten unterstützt jedes Prozessmodell auf seine Weise: Während im RUP der Einzelne seine Erkenntnisse anderen vor allem in Form von Dokumenten verfügbar macht, tauscht er sich in XP mündlich mit anderen aus.

Da beide Ansätze wertvolle Beiträge bezüglich den Herausforderungen des Wissensmanagements leisten, wird empfohlen, von beiden zu profitieren. Im Softwareprozess eines Unternehmens sollte eine geeignete Mischung zwischen Kommunikations- und Dokumentenorientierung gefunden werden. So kann einerseits sichergestellt werden, dass informelle Netze entstehen und implizites Wissen ausgetauscht wird. Andererseits wird der Gefahr des organisationalen Vergessens entgegengewirkt, indem Dokumente verfasst werden, welche das Wissen aus den Köpfen der Mitarbeiter zumindest teilweise abbildet.

Quellenverzeichnis

Albrecht, Karl (2003): The Power of Minds at Work – Organizational Intelligence in Action. Amacom, New York 2003.

Balzert, Helmut (1998): Lehrbuch der Softwaretechnik Bd. 1 - Software-Entwicklung. Spektrum Akademischer Verlag, Heidelberg, Berlin 1998.

Barth, Steve (2002): Defining Knowledge Management. http://www.destinationkm.com/articles/default.asp?ArticleID=949, 2002, Anhang D.

Beck, Kent (2000): Extreme Programming Explained – Embrace Change. Addison Wesley, Boston 2000.

Beck, Kent; Fowler, Martin (2000): Planning Extreme Programming. Addison Wesley, Boston 2000.

Bukowitz, Wendi R.; Williams, Ruth L. (2002): Wissensmanagement – Effizientes Knowledge Management aufbauen und integrieren. Financial Times, München 2002.

Cockburn, Alistair; Highsmith, Jim (2001): Agile Software Development: The People Factor. In: Computer, November 2001, 131-133.

De Marco, Tom; Lister, Timothy (1998): Wien wartet auf Dich – Der Faktor Mensch im DV-Management. Carl Hanser Verlag, München & Wien 1998.

De Marco, Tom (2002): Slack – Getting past burnout, busywork, and the myth of total efficiency. Boradway Books, New York 2002.

Duden (1993): Informatik – Ein Sachlexikon für Studium und Praxis, Dudenverlag, Mannheim 1993.

Elssamadisy, Amr: XP on a Large Project – A Developer's View. Ohne Jahresangabe. http://thoughtworks.com/library/XP%20On%20A%20Large%20Project%20-%20A%20Developer%27s%20View.pdf. Anhang I.

Fowler, Martin (1999): Refactoring: Improving the Design of Existing Code. Addison Wesley, Boston 1999.

Friedewald, Michael; Rombach, H. Dieter; Stahl, Petra; Broy, Manfred;Hartkopf Susanne; Kimpeler Simone; Kohler, Kirstin; Wucher, Robert; Zoche, Peter (2001): Sofwareentwicklung in Deutschland – Eine Bestandesaufnahme. In: Informatik Spektrum 24 2001, 81-90.

Giddens, Anthony (1992): Die Konstitution der Gesellschaft – Grundzüge einer Theorie der Strukturierung. Campus Verlag, Frankfurt/Main.

Goleman, Daniel (1996): Emotionale Intelligenz. Carl Hanser Verlag, München 1996.

Gygi, Beat (2002): Versicherungsprämien steigen – Die Versicherungwirtschaft kann es sich nach dem Börsensturz nicht mehr leisten, mit Prämien zu arbeiten, die Kosten und Schäden nicht decken. In: NZZ am Sonntag, 18.8.2002, Nr. 23, S. 31.

Jeffries, Ron; Anderson, Ann; Hendrickson, Chet (2001): Extreme Programming Installed. Addison Wesley, Boston 2001.

Kruchten, Philipp (1996): A Rational Development Process. In: Crosstalk, 9 (7) July 1996, 11 – 16.

Kruchten, Philipp (2000): The Rational Unified Process – An Introduction, 2. Aufl., Addison Wesley, Reading 2000.

Lehner, Franz (2000): Organisational Memory - Konzepte und Systeme für das organisatorische Lernen und das Wissensmanagement. Carl Hanser Verlag, München 2000.

Martin, Robert C. (2001): Agile Processes.
http://www.objectmentor.com/resources/articles/agileProcess.pdf. Anhang E.

Meyer, E. (2003): Abbau von 350 Stellen bei der "Winterthur" – Umfang der Reduktion bei der Credit Suisse ungewiss. In: Neue Zürcher Zeitung, 26.2.2003, Nr. 47, S. 39.

Nonaka, Ikujiro; Takeuchi, Hirotaka (1997): Die Organisation des Wissens - Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen. Campus Verlag, Frankfurt / New York 1997.

North, Klaus (2002): Wissensorientierte Unternehmensführung – Wertschöpfung durch Wissen. Gabler Verlag, Wiesbaden 2002.

Pawlowsky, Peter (1998): Integratives Wissensmanagement. In: Pawlowsky, Peter (Hrsg.): Wissensmanagement: Erfahrungen und Perspektiven. Betriebswirtschaftlicher Verlag Dr. Th. Gabler GmbH, Wiesbaden 1998, 9-46.

Pawlowsky, Peter; Reinhardt, Rüdiger (2002): Instrumente Organisationalen Lernens: Die Verknüpfung zwischen Theorie und Praxis. In: Pawlowsky, Peter; Reinhardt, Rüdiger (Hrsgs.): Wissensmanagement für die Praxis - Methode und Instrumente erfolgreicher Umsetzung, Hermann Luchterhand Verlag GmbH, Neuwied 2002, 1-35.

Pfiffner, Fritz (2002): "Intelligente" Fonds-Policen. In: NZZ am Sonntag, 24.11.2002, Nr. 37, S.66.

Probst, Gilbert J. B.; Knaese Birgit (1998): Risikofaktor Wissen: Wie Banken sich vor Wissensverlusten schützen. Gabler, Wiesbaden 1998.

Probst, Gilbert J. B.; Raub, Steffen; Romhardt, Kai (1999): Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Gabler, Wiesbaden 1999.

Rasonyi, P. (2001): Was die Credit Suisse derzeit umtreibt – Sinn und Risiken der laufenden Restrukturierung. In: Neue Zürcher Zeitung, 22.12.2001, S. 23.

Schneider, Ursula (2001): Die 7 Todsünden im Wissensmanagement – Kardinaltugenden für die Wissensökonomie. Frankfurter Allgemeine, Frankfurt/Main 2001.

Senge, Peter M. (2001): Die fünfte Disziplin – Kunst und Praxis der lernenden Organisation, 8. Aufl., Klett-Cotta, Stuttgart 2001.

Sommerville, Ian (2001): Software Engineering, 6. Aufl., Addison Wesley, Boston 2001.

Versteegen, Gerhard (2000): Projektmanagement mit dem Rational Unified Process. Springer Verlag, Berlin / Heidelberg 2000.

Von Felbert, Dirk (1998): Wissensmanagement in der unternehmerischen Praxis. In: Pawlowsky, Peter (Hrsg.): Wissensmanagement: Erfahrungen und Perspektiven. Betriebswirtschaftlicher Verlag Dr. Th. Gabler GmbH, Wiesbaden 1998, 119-141.

Von Krogh, Georg; Ichijo, Kazuo; Nonaka, Ikujiro (2000): Enabling Knowledge Creation - How to unlock the mystery of tacit knowledge and release the power of innovation. Oxford University Press, New York 2000.

Willke, Helmut (2001): Systemisches Wissensmanagement. Mit Fallstudien von Karsten Krück, 2. Aufl., Lucius & Lucius, Stuttgart 2001.

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Ich habe mich nicht anderer als der im beigefügten Quellenverzeichnis aufgeführten Hilfsmittel bedient. Alle Stellen, die wörtlich oder sinngemäss aus Veröffentlichungen übernommen wurden, sind als solche gekennzeichnet.

Katja Stucki