



Fraunhofer Institut
Experimentelles
Software Engineering

Konzeption und Evaluation von Inspektions- Methoden zur Qualitätssicherung von Erfahrungen und Wissen aus dem Software Engineering

Imane Hanbali

November 2003

Betreuer: Prof. Dr. H. Dieter Rombach
Jörg Rech

Diplomarbeit
Version 1.0

Eine Publikation des Fraunhofer IESE

Erklärung

Hiermit erkläre ich, Imane Hanbali, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Kaiserslautern, den 07 November 2003

Imane Hanbali

Abstract

Diese Diplomarbeit befasst sich mit der Konzeption und Entwicklung von Methoden zur Inspektion von Wissen und Erfahrungen. Dazu müssen zuerst theoretische Grundlagen vorbereitet werden.

Dabei werden die vorhandenen Inspektionsmethoden im Software Engineering gründlich betrachtet. Es werden deren Ziele, Prozesse, Modelle und Rollen vorgestellt. Anschließend wird auf bekannte Prüfverfahren eingegangen.

Gegenstand dieser Diplomarbeit sind die Themen Wissen und Wissensmanagement. Denn bevor man Inspektionsmethoden zu Wissen entwickeln kann, muss dieser Begriff auf naturwissenschaftlicher Weise definiert werden. Zu dem Konzept der Wissensinspektion stellt sich automatisch die Frage nach Wissensmanagement. Daher werden hier einige Konzepte vorgestellt und es wird auf die Grundbausteine von Probst eingegangen.

Nachdem der Weg mit theoretischen Grundlagen aufbereitet wurde, werden in Anlehnung an die Modelle der Inspektionsmethoden im Software Engineering drei Methoden für die Inspektion von Wissen und Erfahrungen entwickelt. Ziel ist dabei die Ad-hoc, die Checklistenbasierte und die Perspektiven basierte Lesetechniken zu verfeinern und an die Wissensdokumente anzupassen, diese werden im Rahmen eines Experimentes dann evaluiert und analysiert. Des weiteren werden aus den erzielten Ergebnissen sogenannte „Lessons Learned“ abgeleitet.

Ein Ausblick darüber wie diese Arbeit auf dem Gebiet der Inspektionsmethoden von Wissen und Erfahrungen dann weitergeführt werden kann, rundet den Abschluss.

Schlagworte: Inspektionsmethoden im Software Engineering, Lesetechniken in der Software, Wissensmanagement, Methoden zur Inspektion von Wissen, Methoden zur Inspektion von Erfahrungen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Software Inspektionen	2
1.2	Wissensinspektionen	3
1.3	Wissen und seine Eigenschaften	3
1.4	Ziele der Diplomarbeit	4
1.5	Nutzen der Arbeit	8
2	Inspektion im Software Engineering	11
2.1	Qualitätssicherung und Inspektionen	11
2.2	Was sind Inspektionen	14
2.3	Inspektionen im Software Engineering	14
2.4	Ziele und qualitative Vorteile einer Inspektion	16
2.5	Der Prozess der Inspektion	18
2.5.1	Der Inspektionsprozess für informelle Prüftechniken	19
2.5.2	Der Inspektionsprozess für formale Prüftechniken	20
2.6	Rollen bei der Inspektion	23
2.7	Das Vorgehensmodell und die Inspektion	25

2.8	Prüfverfahren im Software Engineering	26
2.8.1	Informale Lesetechniken	26
2.8.2	Formale Inspektionsmethoden	27
2.9	Beurteilung von Inspektionstechniken	32
2.10	Modelle der verschiedenen Lesetechniken	33
2.11	Zusammenfassung	36
3	Von Wissen zu Wissensmanagement	39
3.1	Wissen	39
3.2	Der Anreicherungsprozess: Daten zu Information zu Wissen	41
3.3	Wissen in der Epistemologie	44
3.4	Das HANSE-Modell	46
3.5	Wissensakquisition	48
3.6	Wissensträger	49
3.7	Wissensdarstellungsformen	51
3.8	Wissensmanagement	53
3.9	Bausteine des Wissensmanagements	56
3.9.1	Wissensidentifikation:	57
3.9.2	Wissenserwerb:	58
3.9.3	Wissensentwicklung:	58
3.9.4	Wissens(ver)teilung:	58
3.9.5	Wissensnutzung:	59
3.9.6	Wissensbewahrung:	59
3.10	Typen von Wissensmanagement Systeme	60
3.11	Zusammenfassung und Fazit	61
4	Konzeption und Entwicklung der Inspektionsmethoden	64
4.1	Die Konzeption der Inspektionsmethoden für Wissen und Erfahrungen	64
4.1.1	Probleme eines Wissensdokumentes	65
4.1.2	Qualitätsmerkmale eines Wissensdokumentes	65
4.1.3	Vorteile einer Wissensinspektion	67
4.2	Auswahl und Entwicklung der Inspektionsmethoden	69
4.2.1	Auswahl der Methoden	71
4.2.2	Entwicklung der Methoden	71
4.3	Das Experiment	74
4.3.1	Einführung	75

4.3.2	Ad-hoc	80
4.3.3	Checklisten basierte Lesetechnik(CBR)	80
4.3.4	Perspektiven basierte Lesetechnik	82
4.4	Die praktische Inspektion	86
4.5	Evaluation	89
4.6	Zusammenfassung	96
4.6.1	Ad-hoc:	97
4.6.2	Checklisten basierte Lesetechnik:	97
5	Zusammenfassung und Ausblick	100
5.1	Ziele der Arbeit	100
5.2	Experimentelle Beschreibung und Auswertung	103
5.3	Lessons learned	104
5.4	Ausblick	107
	Danksagung	109
	Abbildungsverzeichnis	110
	Tabellenverzeichnis	112
	Literatur	113

1 Einleitung

Im Angesicht des heute immer größer werdenden Zuwachs an Wissensbasierten Systemen und wissensorientierten Organisationen im Allgemeinen, ist aus Sicht erfahrener Software Ingenieure und Fachleuten nur verständlich nach geeigneten Inspektionsmethoden für Wissen und Erfahrungen der heutigen Informatik zuzusehen. Vor allem nach den großen Erfolgen, die die Inspektionsmethoden in der Software Entwicklung erzielt haben.

Wissen ist heutzutage zu einem riesigen Produktionsfaktor geworden. Firmen versuchen durch wissensbasierte Systeme und Wissensmanagement-Effizienz ihre Wettbewerbsfähigkeit Ausdruck zu verleihen, und dadurch sich einen Vorsprung zu ihren Mitbewerbern zu schaffen.

Aus diesem Eifer heraus, sind verschiedene Ideen und Implementierungen von wissensbasierten Systemen entstanden. Sie sind so vielfältig, dass sie daher einer gründlichen Überprüfung unterzogen werden müssen.

Die allererste Frage die sich in den Raum stellt, kommt aus der Tatsache, dass man Wissen und seine Qualität nicht messen kann: *kann man denn Wissen überhaupt inspizieren?*

Diese Frage zieht andere Fragestellungen automatisch mit sich, nämlich:

- *Wie ist Wissen überhaupt definiert?*
- *Hat Wissen eine wohldefinierte Struktur?*
- *Ist Wissen von seinem Kontext unabhängig?*

Um diese Fragen zu beantworten, werden wir Wissen erst mal definieren müssen. Dieses werden wir ausführlich im dritten Kapitel machen. Lassen Sie uns hier aber folgendes festhalten:

Es gibt zwei Hauptkategorien von Wissen:

- **Explizites Wissen:** dies ist jenes Wissen, das immer greifbar ist und in irgend einer Form dokumentiert wird. Sei es auf elektronischer oder herkömmlicher Weise.
- **Implizites Wissen:** diese Form von Wissen wird auch das stille Wissen genannt. Es ist jenes Wissen, das sich meistens in den Köpfen der Experten festsetzt.

Nach diesen beiden Definitionen wollen wir dann festhalten, dass Explizites Wissen sich auf jeden Fall Mithilfe geeigneter Methoden inspizieren lässt.

1.1 Software Inspektionen

In Anlehnung an das erfolgreiche Modell der Inspektionen im Software Engineering soll nun eine Inspektionsmethodik für explizites Wissen konzipiert werden. Diese sind auch vor drei Jahrzehnten aus dem Wunsch heraus entstanden, die Qualität eines Softwareproduktes zu optimieren. Die Inspektionsmethoden haben sich als sehr gute und effektive ständige Qualitätskontrolle erwiesen. Daher werden sie nun von vielen Firmen in den Entwicklungsprozessen miteinbezogen.

Aber wenn wir über Software Qualität sprechen, müssen wir diesen Begriff definieren. Dieser Begriff ist aber nicht leicht zu definieren. Denn Softwarequalität ist von mehreren Faktoren abhängig. Mehrere Eigenschaften einer Software definieren eben ihre Qualität. Diese Qualitätsmerkmale lassen sich wiederum beliebig kombinieren, abhängig von den Benutzeranforderungen oder Herstellerwünschen. Die bestmögliche Qualität eines Software Produktes lässt sich erreichen, in dem man die gewünschten Merkmale dieses Produktes, ausgehend von den entsprechenden Anforderungen, immer wieder inspiziert.

Software hat sich nun weitgehend etabliert. Der Trend geht nun in die Richtung, dass die Organisationen ihre Informationsbestände

wissensbasiert errichten. Dieser Trend bringt die Überlegungen mit sich, ob das Wissen auch qualitativ gut genug ist. D.h. auch die Frage mit sich, welche Strategien lassen sich benutzen, um genau wie auf dem Software Gebiet, eine ständige Kontrolle der Wissensartefakte zu ermöglichen.

1.2 Wissensinspektionen

Die Wissensqualität ist genau wie die Softwarequalität auch von mehreren Faktoren abhängig. Es kommt sogar die Schwierigkeit bei den Wissensdokumenten hinzu, dass das Wissen mehrere Qualitätsmerkmale besitzt, die nicht immer klar zu definieren sind. Diese Schwierigkeit steigt mit der Kombination von diesen Eigenschaften.

Die nächste Frage, die wir hier anschneiden wollen, ist: *was wohl Wissens Elemente sein könnten?*

Das Wissen lässt sich vielfältig definieren. Wissen aus dem Bereich des Software Engineerings sind beispielsweise Projekterfahrungen, Fälle und deren Lösungen aus einer Fallbasis, Berichte, Mitarbeiter Know-how und vieles mehr. Diese Wissens Elemente müssen an sich verschiedene Kriterien erfüllen. In dieser Diplomarbeit wird dieses Thema detaillierter behandelt, so dass die daraus gewonnenen Erkenntnisse in die drei ausgewählten Prüftechniken einfließen können.

1.3 Wissen und seine Eigenschaften

Das zweite Teil dieser Diplomarbeit dient der Vorstellung der wichtigsten heute verfügbaren Methoden im Software Engineering. Diese werden in tiefer Überlegung dann auf ihre Tauglichkeit für Wissensbestandteile studiert. Dazu muss das Thema Wissen intensiv behandelt werden, seine Eigenschaften und seine Komponenten bekannt werden. Es ist schon im Voraus zu erwarten, dass auf diesem Bereich anderen Problemen zu begegnen sind als auf dem Gebiet der Software Inspektionen. Denn, anders als Softwareprodukte sind Wissensprodukte:

- Nicht strukturiert
- Nicht immer verständlich
- Und sehr kontextabhängig

Diese typischen Eigenschaften eines Wissensdokumentes werden sicherlich die Hauptschwierigkeiten einer effektiven und schnellen Inspektion darstellen. Diese Besonderheiten müssen dann bei der Auswahl und Entwicklung von der Inspektionsmethoden während dieser Diplomarbeit in Betracht gezogen werden. In einem Experiment wird dann überprüft, in wie fern sich die ausgewählten Methoden tatsächlich eignen.

1.4 Ziele der Diplomarbeit

Diese Diplomarbeit beinhaltet folgende Schritte:

1. Recherche: zuerst müssen die theoretischen Grundlagen von Inspektionen im Software Engineering vorlegen. Dann müssen die Themen Wissen und Wissensmanagement gründlich behandelt werden.
2. Auswahl der Methodik: aus den wohldefinierten Inspektionsmethoden im Software Engineering werden drei Methoden ausgewählt, die zum Inspizieren von Wissens-elementen geeignet aussehen.
3. Verfeinerung und Anpassung der ausgewählten Prüfverfahren: die drei ausgewählten Methoden werden zu den Merkmalen und Eigenschaften von Wissens-elementen verfeinert und angepasst, unter Berücksichtigung der Problematik von manchen Wissens-eigenschaften.
4. Experiment und Evaluierung: in einem Experiment sollen diese Prüfverfahren durchgeführt werden und evaluiert.

Unsere Ziele bei der Durchführung von Wissensinspektionen dienen alle der Optimierung der Qualität der einzelnen Wissens-elemente, so dass:

- der Benutzer dieser Wissens Elemente bestens bei seinen täglichen Aufgaben unterstützt wird,
- falls ähnliche Aufgaben oder Projekte wieder auftreten diese schneller bearbeitet werden können und
- die Fähigkeiten des Benutzers selbst verbessert werden, in dem er von diesen Wissens Elementen lernt, und von der Art und Weise wie ein vergangenes Problem gelöst wurde, besonders bei Problem/Lösungsfällen.

Diese erwünschte Optimierung der Qualität von Wissensdokumenten wird stark von den ausgewählten Inspektionsmethoden abhängen. Dies ist aus den Erfahrungen der Inspektionsmethoden im Software Engineering abzuleiten. Aus diesem Grund werde ich die meisten heutzutage bekannten und benutzten Inspektionsmethoden der Software Engineering auch auf ihre Effizienz überprüfen.

Aus den Konzepten der einzelnen Inspektionsmethoden im Software Engineering und aus den Erkenntnissen vieler Fachleuten auf diesem Gebiet, ist hier die Frage zu beantworten, welche Methoden sich nun am besten zur Durchführung von sinnvollen Inspektionen an Wissens Elementen eignen. Die Aufgabe dabei ist, drei Methoden auszuwählen, diese weiterzuverfeinern, sodass sie den angestrebten Qualitätsmerkmalen eines Wissensdokumentes am ehesten erfüllen. D.h. diese Inspektionsmethoden müssen die schwierigeren Eigenschaften eines Wissensdokumentes im Allgemeinen berücksichtigen, leicht verständlich sein für potentielle Inspektoren, gut durchführbar sein, und effektive Ergebnisse erzielen können.

Nach gründlicher Anschauung der meisten im Software Engineering verfügbaren Prüfverfahren, schienen die Konzepte der folgenden Softwareinspektion am besten geeignet:

- Ad-hoc, da die Ad-hoc Inspektion einfach und überall einsetzbar ist. Sie verlangt von dem zu inspizierenden Dokument keinerlei Struktur und auch keinen Formalismus. Allerdings bietet sie den Inspektoren keinerlei Unterstützung.

- Checklisten basierte Lesetechnik. Diese Methode eignet sich auch zu unseren Zwecken sehr gut, da den Wissensinspektoren mehr Unterstützung geboten wird, in Form von Checklisten, die an alle Inspektoren verteilt werden. Diese Checklisten beinhalten geschlossene maßgeschneiderte Fragen, die speziell auf das Wissensselement ausformuliert werden. Auf dieser Weise erhalten die Inspektoren einen Leitfaden, der sie bei der Überprüfung unterstützt.
- Die Perspektiven basierte Lesetechnik bietet die Möglichkeit an, das Wissensselement aus verschiedenen Sichten zu überprüfen. Sie sieht Rollen vor für die Inspektoren und definiert individuell für jede Rolle eine Checkliste. Sie eignet sich nach gründlicher Betrachtung zur Inspektion von Wissensartefakten auch sehr gut aus folgenden Gründen.

Die Perspektiven basierte Lesetechnik erzielt die meiste Effektivität nach den Publikationen der Experimente und Evaluation, die von verschiedenen Software Experten durchgeführt wurden. Deshalb habe ich mir auch zum Ziel gesetzt, diese Methode für die Zwecke dieser Diplomarbeit anzupassen, und sie in einer adaptierten Form beim Experiment einzusetzen. Dabei ist es erwünscht, ein Wissensdokument von verschiedenen Blickwinkeln betrachten und überprüfen zu lassen, um die Anzahl der entdeckten Fehler zu steigern. Diese Idee eignet sich aus unserer Sicht auch gerade zu der Idee der ganzheitlichen Wissensstruktur und Organisation, da diese Inspektionsmethode auch selbst das ganzheitliche Prinzip verfolgt. Denn durch die verschiedenen Sichten werden auch verschiedene Ursachen entdeckt. Ein weiterer Vorteil dieser Technik ist, dass das Wissensselement nicht nur aus verschiedenen Sichten inspiziert werden kann, sondern aus verschiedenen Sichten verschiedener Stakeholder, was unter der Stakeholder-Analysis bekannt ist. Im Laufe der Überlegung wie diese Perspektiven zu definieren sind, wurde folgende Strategie verfolgt: Die Rollen hängen hauptsächlich davon ab, welche notwendigen Eigenschaften eines Wissensdokumentes unabdingbar überprüft werden müssen. Die Wissensartefakte werden dann aus jeder sinnvollen Perspektive betrachtet und inspiziert, was zur Fehlerfindung beitragen könnte. Danach werden die Perspektiven definiert. Dazu müssen diese operationalen Szenarien, die die

Beschreibung einer oder mehrerer Aktivitäten beschreiben, und eine Menge von Fragen beinhalten, entwickelt werden. Sie werden eine Anleitung für die Gutachter wie die Dokumente aus der entsprechenden Perspektive zu lesen sind beinhalten, eine Menge von Aktivitäten und eine dazu entsprechende Menge von Fragen. Dabei können nur Fehler entdeckt werden, die für die vorgegebene Sicht relevant sind. Bei dieser Methode wird erhofft in dem Experiment die höchste Effizienz und Fehlerfindung zu erzielen. Da die Problematik bei den Wissens-elementen um einiges größer als bei den Software Dokumenten, wird Mithilfe verschiedener Perspektiven versucht, in das unstrukturierte Wissensdokument etwas Struktur reinzubringen. Denn meistens sind die Wissens-elemente keiner Struktur unterlegen

Ein Bestandteil dieser Diplomarbeit ist die Durchführung eines Inspektionsexperimentes mit einem Inspektorenteam. Die erzielten Ergebnisse werden dann auch gesondert nach der Evaluation betrachtet. Anhand der Evaluation wird dann festgestellt, welche Methode sich bei der Inspektion von Wissensdokumenten tatsächlich bewährt hat.

In diesem Experiment werden die für unsere Zwecke weiterentwickelten und verfeinerten drei Inspektionsmethoden durchgeführt. Dabei werden verschiedene Wissensdokumente inspiziert. Das Experiment wird mit den Studenten des Hauptstudiums der Vorlesung „ERFAHRUNGSMANAGEMENT – ENTWICKLUNG UND EINFÜHRUNG“ von Dr. Althoff durchgeführt. Die Dokumente, die hier getestet werden reichen von internen Wissensdokumente wie Projekterfahrungen und Problem Lösungspaare des „Experience Base Demonstrator“ bis hin zu Dokumenten des Visek-Projektes und Defektfällen von Bugzilla, das Defekt Management System von Mozilla.

Die Studenten werden Gelegenheit haben, sich mit den Prüftechniken auseinander zu setzen, in dem sie gründlich eingeführt werden und die verschiedenen Methoden mit ihren Vorteilen und Nachteilen kennen lernen. Außerdem erlernen dabei das praktische Testen und dadurch Erfahrung mit konkreten Wissensdokumenten und den Prüftechniken erhalten.

Die Ergebnisse und das von den Studenten erhaltene „Feedback“ werden dann aufgearbeitet und zu „Lessons learned“ umgewandelt werden, und somit zur Weiterentwicklung von Inspektionsmethoden für Wissen und Erfahrungen beitragen.

Das Experiment selbst ist natürlich heißerwartet aber auch die Ergebnisse und die Informationen, welche die Evaluation an das Tageslicht bringen würde.

1.5 Nutzen der Arbeit

Mit dieser Arbeit wird erhofft, ein Stück Wissenschaftsarbeit zu liefern, das dazu beitragen wird, Erkenntnisse zu gewinnen, wie sinnvoll eine Wissensinspektion sein kann. Dass daraus Kenntnisse entnommen werden können, die zur Etablierung von Wissensinspektionen in heutigen Wissensorganisationen vorantreiben, sodass sich die Wissensinspektionen sich verbreiten und bewähren wie heute die Softwareinspektionen.

Denn durch die Anwendung der Inspektionsmethoden an den verschiedenen Wissens-elementen seitens einer Wissensorganisationen, soll eine sichtbare Optimierung der Qualität erreicht werden und damit einen für die Firmen spürbaren Nutzen auf drei Ebenen:

- Die technische Perspektive sieht den Nutzen darin, dass eine bessere Explikation von Wissen dadurch erzielt werden kann. Dies drückt sich aus in der verbesserten Wiederverwendung von Technologien und die verbesserte Anwendung der Lessons Learned.
- Ökonomische Perspektive: Durch die ständige Kontrolle der Wissens-elemente, wird sich die Qualität des Wissens von Organisationen und sogar der Mitarbeiter auf jeden Fall positiv ändern. Dies führt unmittelbar zum Unternehmenserfolg. Denn allein die Verkürzung der Entscheidungszeit bei den Mitarbeitern führt direkt zu einer Erhöhung der Produktivität des Unternehmens selber.

- Aus der Soziologisch-organisatorischen Perspektive wird sogar die Verbesserung der Arbeitsbedingungen der Mitarbeiter erwünscht und tatsächlich auch erzielt [Pro97].

Aufbau dieser Diplomarbeit

Diese Diplomarbeit ist in fünf Kapiteln gegliedert. Das erste Kapitel, die Einleitung, befasst sich u.a. mit der Motivation, die zur Entstehung dieser Diplomarbeit geführt hat. Sie gibt einen detaillierten Überblick über die gesamte Arbeit, die Problematik die im Moment bei den Wissensdokumenten zu finden ist, und beschreibt wie mit dieser Arbeit einen Beitrag zur Inspektion von Wissen und Erfahrungen geleistet werden will.

Das zweite Kapitel legt die ersten Bausteine des benötigten theoretischen Hintergrunds für diese Diplomarbeit. Es behandelt nämlich die Entstehung der Inspektionsmethoden im Software Engineering, ihren Zusammenhang mit der Qualitätssicherung. Es definiert verschiedene Inspektionsprozesse und alle darin vorkommenden Rollen und stellt außerdem die meisten zur Zeit bekannten Inspektionsmethoden im Software Engineering vor.

Das dritte Kapitel befasst sich mit dem Thema Wissen und dessen Managements. Es behandelt die Wissensformen, ihre Definitionen auf den verschiedenen Ebenen. Es gibt noch dazu eine detaillierte Einführung in das Wissensmanagement und stellt verschiedene Wissensmanagementtypen vor.

Das vierte Kapitel befasst sich mit der Konzeption und Entwicklung der Wissensinspektionsmethoden. Hier werden dann die in dieser Diplomarbeit entwickelten Methoden vorgestellt. Es werden die Gründe beschrieben, die ausschlaggebend für die Auswahl der einzelnen Inspektionsmethoden waren. Es wird außerdem den Rahmen des Experimentes ausführlich beschrieben. Außerdem befasst sich dieses Kapitel mit der Evaluierung des Experimentes. Es

werden hier die Erkenntnisse aus dem Experiment dann vorgestellt.
Das Kapitel schließt dann mit einer Zusammenfassung zu den
einzelnen Inspektionen.

Das Abschlusskapitel dient die Zusammenfassung der Arbeit und gibt
den Ausblick dieser Diplomarbeit.

2 Inspektion im Software Engineering

In diesem Kapitel geht es um Inspektionen im Software Engineering. Das Kapitel beschreibt zuerst den Zusammenhang zwischen Inspektionen und Qualitätssicherung. Dann werden Inspektionen und deren Prozesse allgemein behandelt. Inspektionstechniken werden klassifiziert, und wir werden sehen wie Inspektionen ablaufen, was für Ziele sie haben, und wer sie durchführen sollte. Außerdem werden wir die bekanntesten Inspektionsmethoden vorstellen und beschreiben. Anschließend werden zwei Kriterien zur Beurteilung von Inspektionsmethoden vorgestellt und die Modelle, an denen sich Inspektionen orientieren.

2.1 Qualitätssicherung und Inspektionen

Software wird heutzutage überall eingesetzt. Sie dringt immer mehr auch in die kritischen Bereiche des menschlichen Lebens, u.a. in den Verkehrsbereich und in den medizinischen Bereich. Da Software aber von Menschen entwickelt wird, und Menschen fehlerbar sind, sind Fehler bei Softwaresystemen unumgänglich. Diese Fehler haben verschiedene Auswirkungen bei ihrem Auftreten. Und zwar hängt das Ausmaß dieser möglichen Fehler davon ab, wie Zeit bzw. sicherheitskritisch die Systeme sind.

Die bekanntesten Softwarefehler reichen von erheblich großen materiellen Schäden, bis hin zu menschlichen Opfern. Hierzu folgende Beispiele:

- Elektrische Bestrahlung in Kanada: Im Jahre 1985/86 starben vier Patienten an den Folgen der Fehlbestrahlung mit dem computergesteuerten Strahlentherapiegerät Therac-25. Die Ursachen lagen zum einen an einer Fehleingabe des Bedienpersonals, und zum anderen an der fehlerbehafteten Software des Gerätes. Die Tatsache, dass die Programmierung des Therac-25 eine Soloprogrammierung war, und noch dass der Entwickler auch Selbst das Testen des Gesamtsystems

durchgeführt hat, hat zu den gravierenden Mängeln bei diesem System geführt. Es wurden keine wohldefinierten Testverfahren benutzt, und es gab auch keine ausreichenden Entwicklungs- und Testdokumentation [Mül99].

- Der Schaden, der durch die Explosion der Rakete Ariane 5 im Jahre 1996 betrug eine Milliarde Mark und wurde auch durch einen Softwarefehler verursacht. Dieser hätte vermieden werden können, wenn ausreichende Komponententests durchgeführt worden wären und einen entsprechenden Integrationstest, die im Software Engineering wohldefiniert sind [Rom01].
- Auch bei technischen Systemen spielen Softwarefehler eine große Rolle. Dazu ein aktueller materieller Schaden, der uns die Einführung der LKW-Maut in Deutschland bietet. Diese wurde mehrmals wegen Softwarefehler verschoben. Als die Installation der Geräte schließlich erfolgte, stellte sich heraus, dass diese mit erheblichen Software Defekten behaftet waren. Die Kosten, die dadurch entstanden sind enorm. Allein dem Staat gehen 160 Millionen Mauteinnahmen pro Monat verloren. Das Betreiberkonsortium Toll Collect in Berlin muss mehr als 20.000 Geräte austauschen wegen technischen Fehlern. Eingehende Softwareprüfverfahren hätten dieses Skandal vermieden, bzw. den Ausmaß drastisch reduziert [Ya03] [Yb03] [Ver03].

Diese Missglücke zur heutigen Zeit und in der Geschichte der Software zeigen, welche entscheidende Rolle Qualität in der Herstellung von Software spielt. Die Anforderung bei solchen kritischen Systemen ist, die angestrebte Qualität zu sichern. Die ständige Kontrolle der erzeugten Dokumente spielt eine zentrale Rolle bei der Entwicklung von Software Systemen. Bei jedem Software Produkt muss ständig geprüft werden, ob es seinen Spezifikationen entspricht. Diese Kontrollmaßnahmen nennen wir Prüfstrategien. Sie bilden die grundlegenden Techniken zur Softwarequalitätssicherung und sollen möglichst parallel zum Entwicklungsprozess stattfinden. Qualifizierte Leute überprüfen zu bestimmten Zeitpunkten, ob das Software Artefakt die gewünschte

Qualität erreicht hat oder nicht. Dabei werden Qualitätsdefizite entdeckt. Das Ziel dabei ist, diese Qualitätsmängel bzw. –Defizite in den frühen Stadien des Entwicklungsprozesses zu entdecken, d.h. im Entwurfsstadium oder nach dem Erarbeiten der Spezifikation. Denn ein Fehler ist um so billiger, je früher er entdeckt und beseitigt wird. Ein wesentlicher Vorteil ist auch, dass das Finden von Fehlern in den Frühstadien der Softwareentwicklung die Entstehung von Folgefehler ausschließt. Dies trägt zur Einsparung von Zeit und Geld bei.

Folgende Abbildung gibt uns einen Überblick über die Fehlerkosten in den verschiedensten Stadien [Lig02]

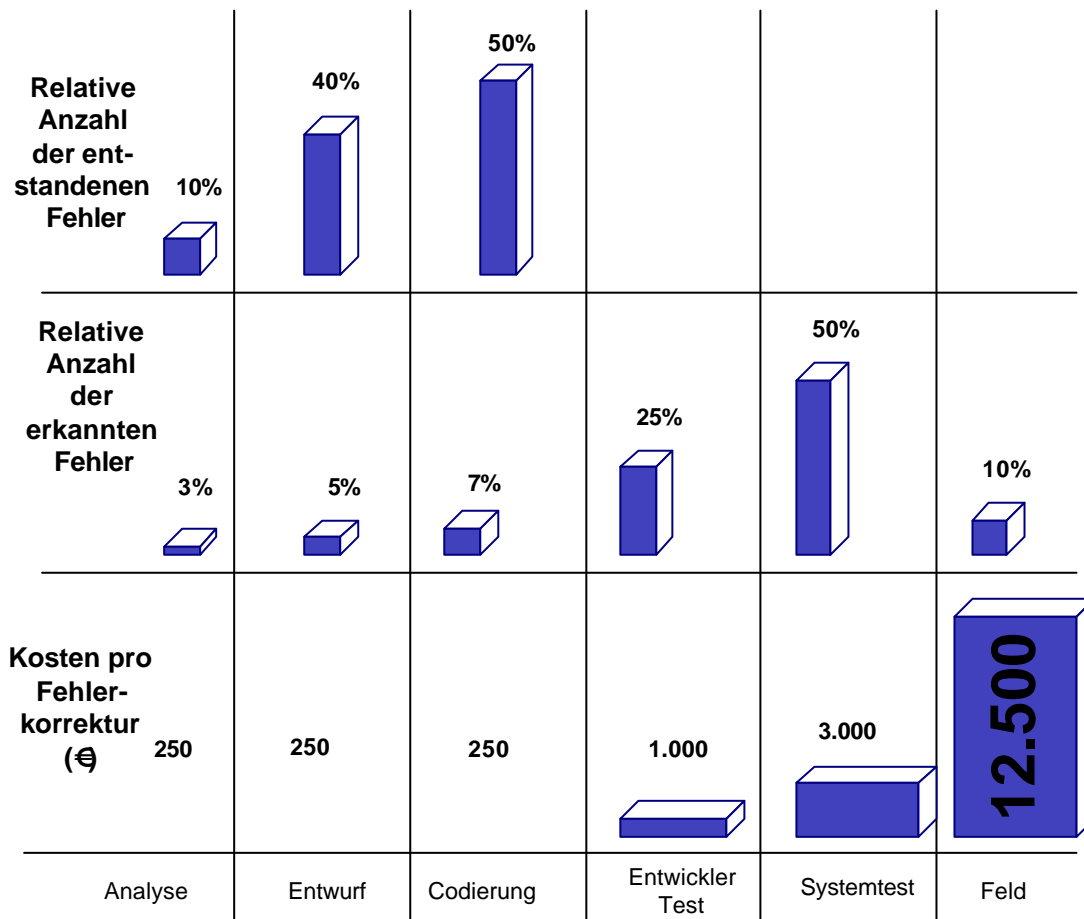


Abbildung 1 Fehleranzahlen und Fehlerkorrekturkosten

Die Bemühung um die Sicherheit von Menschen und ihrer Umgebung, und auch nicht zuletzt die Bemühung um kostengünstige Softwareentwicklung und Wartung, haben dazu geführt, Inspektionen weiterzuführen und neue Ansätze zu entwickeln. So findet man Verfahren speziell für Design, andere nur für Code, oder auch welche die überall einsetzbar sind.

Im nächsten Abschnitt werden wir allgemein mit dem Thema der Inspektion, und danach werden wir einige Inspektionsklassen detaillierter einführen.

2.2 Was sind Inspektionen

Inspektionen gehören zu den manuellen Prüfungen zur Qualitätssicherung und Fehlerreduktion von Software. Inspektionen wurden schon 1976 von Michael Fagan [Fag76] bei IBM entwickelt, um die Software Qualität zu verbessern, und die Produktivität des Programmierers zu steigern. Seitdem haben sich Inspektionen dazu vorgearbeitet, zu dem was sie heute sind, nämlich zu den standhaften, effektiven Methoden zur Qualitätsbesserung und zur Fehlerfindung und Behebung in den verschiedenen Software Artefakten, sobald diese erzeugt werden. Inspektionen aller Art sind unumgänglich, wenn es darum geht, qualitativ hochwertige Software zu entwickeln und zu warten. Sie begleiten die Softwareprodukte im gesamten Lebenszyklus des Softwareprozesses, sprich von der Keimidee bis zum fertigen Produkt, und sogar darüber hinaus, nämlich bei der Wartung.

2.3 Inspektionen im Software Engineering

Im Software Engineering werden Inspektionen wie folgt definiert:

- Im Jahre 1983 definierte IEEE Software Inspektion wie folgt: Software Inspektion (Software Inspection) ist eine formale Bewertungstechnik, die Softwareanforderungen, -Design und -Quellcode genau auf Fehler, Verstöße gegen Entwicklungsstandards und andere Probleme durch eine

Person oder Gruppen, die nicht der Autor selber ist, untersucht [IEEE].

- Im Software Engineering werden Inspektionen als Wohldefinierte und disziplinierte Prozesse definiert. In diesen Prozessen werden Prüftechniken angewandt, die in dynamische und statische Strategien unterteilt werden. Jede dieser Hauptklassen enthält Selbst Unterklassen. Wir werden uns hier mit einer groben Klassifikation begnügen. Wer die genaue Klassifikation wissen möchte, dem empfehle ich das Buch von Liggesmeyer [Ligg2002].

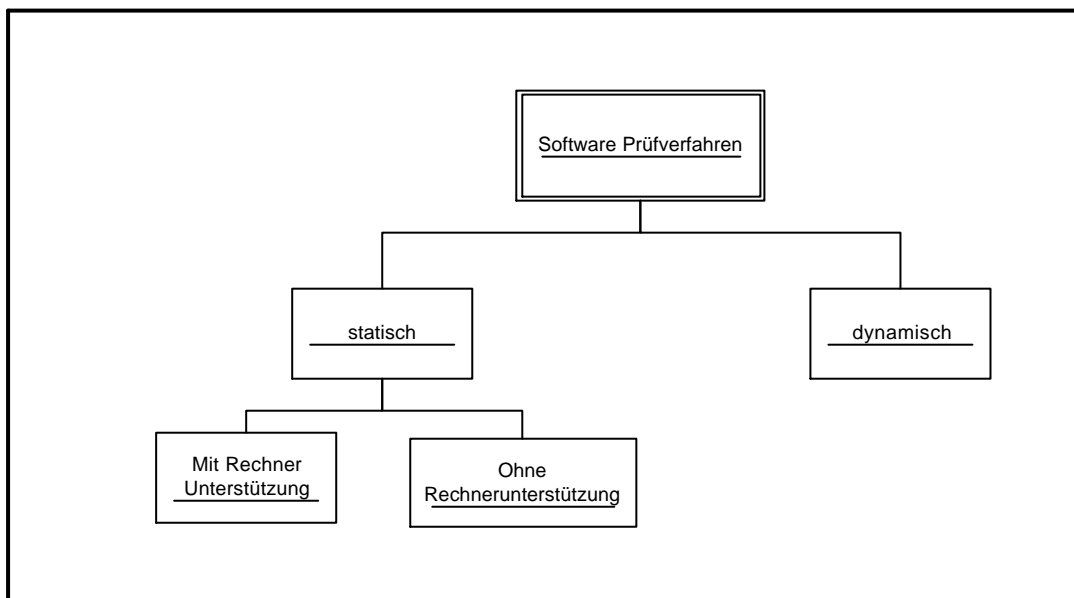


Abbildung 2 Klassifikation von Software Prüfverfahren

Softwareprüfungen werden also in statische und dynamische Verfahren unterteilt. Die dynamischen Inspektionen werden mit Unterstützung verschiedener Softwarewerkzeugen durchgeführt und verfolgen verschiedene Ansätze, d.h. ob sie datenflussorientiert sind oder eher funktionsorientiert. Im Gegensatz dazu werden die statischen Verfahren ohne jegliche Werkzeugunterstützung durchgeführt und kennen auch eine Klassifikation. Diese gibt u.a. formale und informelle Verfahren an. Wir werden die Techniken, die zu diesen beiden Klassen gehören, später kennen lernen.

Inspektionen selbst gehören zu den statischen Prüfverfahren. Es sind formale oder informelle Methoden der Qualitätssicherung, die während des gesamten Lebenszyklus eines Software Produktes eingesetzt werden können. Sie werden eingesetzt, um das Produkt auf die Einhaltung von Standards und die Erfüllung von Produkteigenen Spezifikationen zu überprüfen.

Eine Inspektion kann an allen möglichen Produkten und Teilprodukten durchgeführt werden, in allen Phasen eines Projektes, sobald der Autor, Manager oder Entwickler dies für nötig hält. Die Produkte bzw. Teilprodukte können Anforderungen, Klassendiagramme, Grobentwürfe, Codefragmente, Abnahmeberichte oder Projektpläne sein [Ehr02]. Die Inspektion ist weder an einem Zeitpunkt noch an einem bestimmten Format beschränkt.

Es gibt eine Reihe formaler und informeller Prüfstrategien. In den nächsten Abschnitten werden wir die bekanntesten kennen lernen. Dazu zählen Software Inspektionen, (Peer) Reviews und Walkthroughs. Sowohl formale Inspektionen, (Peer) Reviews als auch Walkthroughs sind statische Methoden zur Softwarequalitätsverbesserung und Fehlerkostenreduktion. Statisch deswegen, da ein Softwaredokument von einem oder mehreren Inspektoren auf Fehler oder Mängel untersucht wird, ohne jegliche Einsetzung von Softwarewerkzeugen.

Das Lesen eines Dokumentes bedeutet, ein Dokument systematisch zu untersuchen, um daraus die nötigen Informationen zu extrahieren, zu gewinnen und die inspizierte Software besser zu verstehen, um dann jegliche Abweichung von gegebenen Standards oder Spezifikationen besser lokalisieren zu können.

In den letzten Jahren haben sich die Inspektionsmethoden mehr und mehr als effektive Lesetechniken durchgesetzt. Sie sind nun zu einem festen Bestandteil des verifikations- und Validationsarsenals geworden. Inspektionen sind dafür geeignet, den ganzen Entwicklungsprozess eines Softwareproduktes zu begleiten.

2.4 Ziele und qualitative Vorteile einer Inspektion

Inspektionen werden aus gewissen Zielen heraus durchgeführt. Ihre Durchführung führt zu sichtbaren Vorteilen.

Die Ziele einer Inspektion sind:

- Defekte der Software Produkte entdecken, identifizieren und beschreiben zu können.
- Daten über den Software Engineering Prozess zu sammeln, beispielsweise Daten über Defekte und Aufwand.
- Die Wiederverwendbarkeit des Software Produktes zu überprüfen.
- Ein delegierter Kontrollmechanismus bereitzustellen, der den nächsten Schritt im Prozess festlegen soll, d.h. Produkt nachbearbeiten, wegwerfen oder in die nächste Phase übernehmen.
- Verbesserung der Qualität der Software Produkte
- Sichtbare Fehlerreduktion zu erzielen
- Eine deutliche Kosteneinsparung zu erreichen

Inspektionen erfreuen sie sich wachsender Beliebtheit, da viele Untersuchungen abhängiger und unabhängiger Institutionen belegen, dass frühzeitige Inspektionseinsetzung dazu führt, 50% bis 90% aller Fehler zu entdecken und zu korrigieren [Lig02]. Somit gelten sie als effektive Methode zur Findung von vielen Fehlern.

Die qualitativen Vorteile von Inspektionen sind vielfältig. Die Einsetzung von Inspektionen aller Art führt zu [Tra96]:

- Der frühen Entdeckung von Fehlern,
- Vereinfachung der Programme,
- Vereinheitlichung des Programmierstils,
- Einer transparenten Software Entwicklung,

- Zuverlässigerer Termin- und Aufwandsschätzung,
- Erhöhter Lerneffekt,
- Verbesserung der Dokumentation und Einhaltung von Standards,
- Erhöhter Zufriedenheit des Benutzers.

2.5 Der Prozess der Inspektion

Aus der Recherche lässt sich folgendes ableiten:

Der Prozess der Inspektion besteht im allgemeinen aus den folgenden fünf Elementen:

- Wohldefinierte Inspektionsschritte
- Wohldefinierte Rollen für die am Inspektionsprozess Teilnehmer (z.B. Moderator, Autor, Inspektor, Schreiber)
- Formale Kollektion von Produkt- und Prozessdaten
- Das zu inspizierende Produkt
- Eine unterstützende Infrastruktur

Der letzte Punkt umfasst organisatorische und Managementorientierte Aspekte einer Inspektion und ist somit nicht Gegenstand dieser Diplomarbeit.

Die zu inspizierenden Produkte können alle Softwareprodukte im Laufe eines Prozesses sein, seien es Dokumentationen, Quelltexte oder Anforderungen aller Art sein, aber auch ein Projektplan selbst kann Gegenstand einer Inspektion sein.

Die Inspektionsschritte und die Rollen werden wir aber genauer in den nächsten beiden Abschnitten betrachten.

Der Prozess der Inspektion wird in Phasen geteilt. Es gibt eine grobe Unterteilung für die informellen Prüftechniken und eine feinere für die formalen Inspektionen.

2.5.1 Der Inspektionsprozess für informelle Prüftechniken

Die grobe Aufteilung ist in drei Phasen gegliedert: Vorbereitungsphase, Durchführung der Inspektion oder die Inspektionssitzung, und Nachbearbeitungsphase.

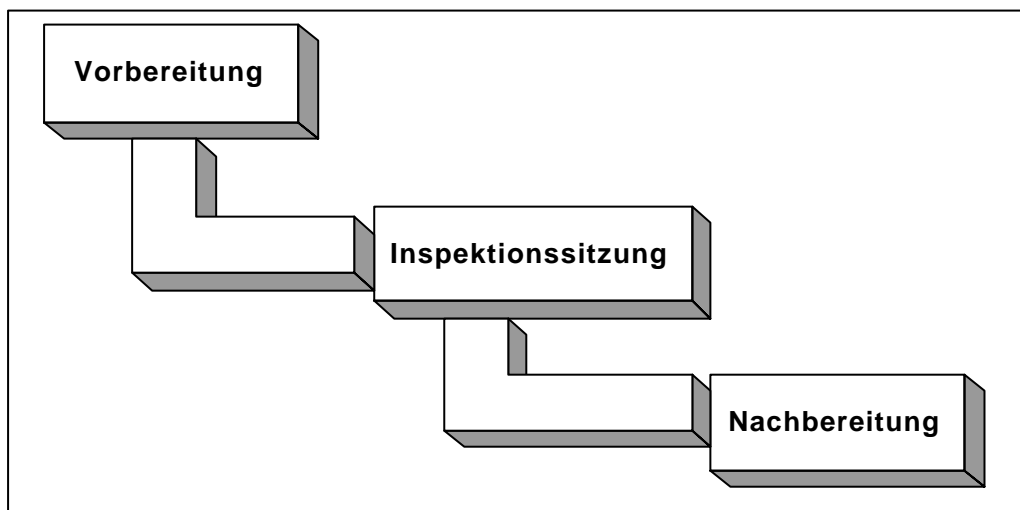


Abbildung3 die drei Phasen einer Inspektion

In der **Vorbereitungsphase** werden die Inspektoren ausgewählt, die Rollen verteilt, das zu inspizierende Artefakt vorgestellt und schließlich einen Termin verabredet für die eigentliche Sitzung, falls eine Gruppensitzung vorgesehen ist.

In der **Inspektionssitzung** wird nach der ausgewählten Inspektionsmethode vorgegangen und Fehler gesucht. Diese kann entsprechend der Prüftechnik entweder in Gruppen durchgeführt werden, oder individuell am Schreibtisch. Die gefundenen Fehler werden dann gesammelt, dokumentiert und an den Autor weitergeleitet.

Schließlich kommt es zu einer **Nachbearbeitung**. Hier wird erst mal nachkontrolliert, ob die gefundenen und dokumentierten Fehler vom

Autor überarbeitet wurden. Überarbeiten kann korrigieren der Fehler sein, oder eine Erklärung warum dieser Schritt genauso durchgeführt werden muss. Schließlich wird vom Management entschieden, ob in die nächste Phase des Projektes gegangen werden darf, oder eine Überarbeitung der aktuellen Phase notwendig sei, aufgrund der gravierenden gefundenen Fehler.

2.5.2 Der Inspektionsprozess für formale Prüftechniken

Die feinere Aufteilung der Inspektion sieht dagegen sechs Phasen vor [Lai00]. Diese sind wie folgt definiert:

1. Die **Planung**: Organisatorische Vorbereitung
2. Die **Überblickssitzung**: Informationsveranstaltung über das Produkt
3. Die **Vorbereitungsphase**: getrennte Vorbereitung der Inspektoren
4. Die **Inspektionssitzung**: Durchführung der Inspektionstechnik in Gruppen
5. Die **Fehlerkorrektur**: Korrektur der Fehler
6. Die **Nachbearbeitung(Follow-up)**: Überprüfung und Dokumentierung der Fehlerkorrekturen

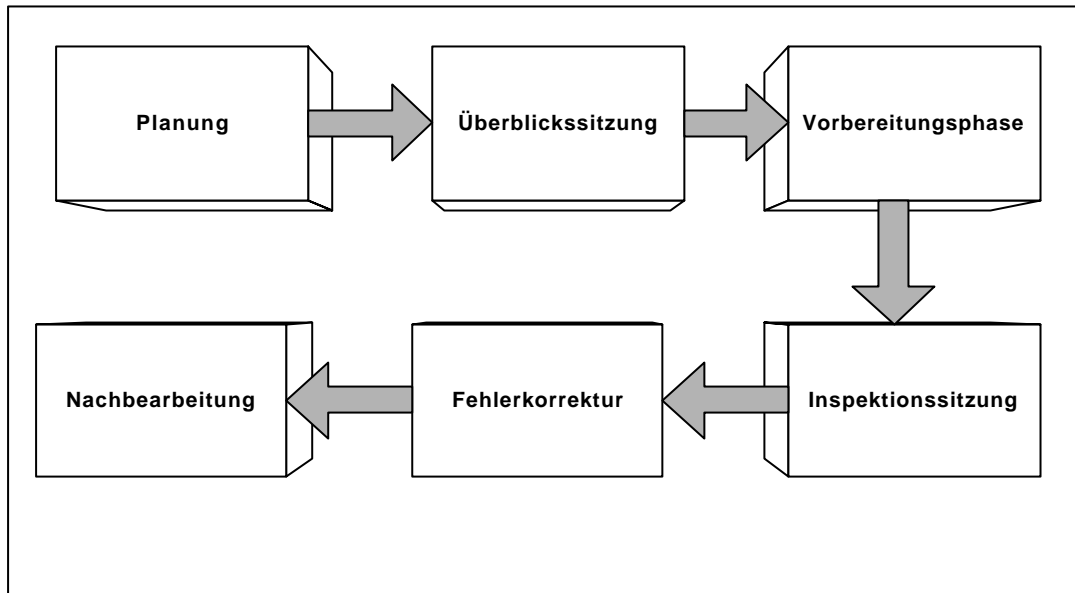


Abbildung4 die sechs Phasen einer Inspektion

In der **Planung** wird die vorgesehene Inspektion organisiert, wenn das Software Artefakt dazu reif ist, z.B. wenn ein Anforderungsdokument fertig erstellt worden ist. Die Entscheidung darüber, ob ein Softwareartefakt reif ist oder nicht, fällt der Moderator. Es wird ein Zeit- Aufwand und Ressourcenplan erstellt. Die Organisation beinhaltet wie oben in der groben Struktur auch der Fall ist, die Aufstellung des Inspektionsteams, die Rollenaufteilung, die Terminplanung einer oder mehreren Inspektionssitzungen und die Auf- und Verteilung des Inspektionsmaterials.

Die **Übersichtsphase** hat zum Hauptziel, das Software Produkt so zu erläutern, dass es für die Inspektoren klar ist, und ihnen die Inspektion erleichtert. Dies geschieht in einem ersten Treffen, in dem der Autor sein Produkt den Inspektionsmitgliedern vorstellt. Da diese Phase allerdings zeitaufwendig ist, wird sie meistens nur in Fällen durchgeführt, wenn das zu inspizierende Material entweder so schwierig oder komplex ist, dass die Autorhilfe unabdingbar ist. Oder wenn das Software Artefakt Teil eines großen Software Systems, und das Verständnis der Beziehung zwischen dem Artefakt und dem gesamten System dadurch erleichtert werden kann. Oder wenn neue Inspektionsmitglieder zum Team erteilt werden. Sie ist auch sinnvoll, wenn eine neue Technik verwendet werden soll. Unter diesen

Umständen scheint diese Übersichtsphase Zeit- und kosteneffektiv zu sein. Allerdings muss Sinnvollerweise darauf geachtet werden, diese Sitzung zeitlich zu begrenzen.

Die **Vorbereitung**: In dieser Phase wird das Software Artefakt auf alle möglichen Fehler gründlich untersucht. Diese Aktivität kann je nach Lesetechnik, entweder in Gruppen durchgeführt werden, oder individuell am eigenen Schreibtisch. Jeder Inspektor notiert jeden Fehler und jede Unklarheit. Meistens bereiten sich die Teilnehmer alleine auf die Sitzung vor, um dann in der Gruppe das Produkt gemeinsam durchzugehen. Welche Variante letztendlich effektiver ist, hängt stark mit den Inspektoren, deren Fähigkeiten und Erfahrungen ab.

Die **Inspektionssitzung** ist der wichtigste Teil einer Inspektion. Hier kommen alle Inspektionsteilnehmer zusammen: Moderator, Autor, Leser, Protokollführer, und die Inspektoren. Sie dient dazu, die von den Inspektionsteilnehmern gefundenen Fehler zu sammeln und zu dokumentieren. Darüber hinaus, wird in dieser Phase darüber entschieden, ob ein Fehler wirklich als Fehler oder nur als Mangel angegeben werden kann, und ob das inspizierte Artefakt noch einmal inspiziert werden soll. Dies alles geschieht in der gesamten Gruppe, oder nur in einer drei Personen Sitzung von Moderator, Autor und dem erfahreneren Inspektors.

In der **Fehlerkorrektur** überarbeitet der Autor die Fehler, und korrigiert die gefundenen Fehler oder rationalisiert ihre Existenz, in dem er Fehler für Fehler die komplette Liste durchgeht.

Die **Nachbearbeitungsphase** bezeichnet man im Englischen mit Follow-up. Das Hauptziel dabei ist festzustellen, ob der Autor alle Defekte überarbeitet hat oder nicht. Diese Überprüfung kann ein Inspektionsmitglied übernehmen. Diese Phase des Inspektionsprozesses bleibt für viele optional, so dass sie meistens nicht durchgeführt wird.

Diese sechs Prozessphasen definieren eine gründliche Inspektion. Einige sind optional wie die Überblicks- und Nachbearbeitungsphase. Welche Phase benutzt wird und in welcher Form sie durchgeführt

wird, zeichnen die Unterschiede zwischen einer Inspektionsmethode und einer anderen aus.

2.6 Rollen bei der Inspektion

Bei einer Inspektion werden Rollen und entsprechende Verantwortlichkeiten unter den Inspektionsteilnehmern aufgeteilt. Die Rollen sind in allen Inspektionsmethoden gleich und werden wie folgt definiert:

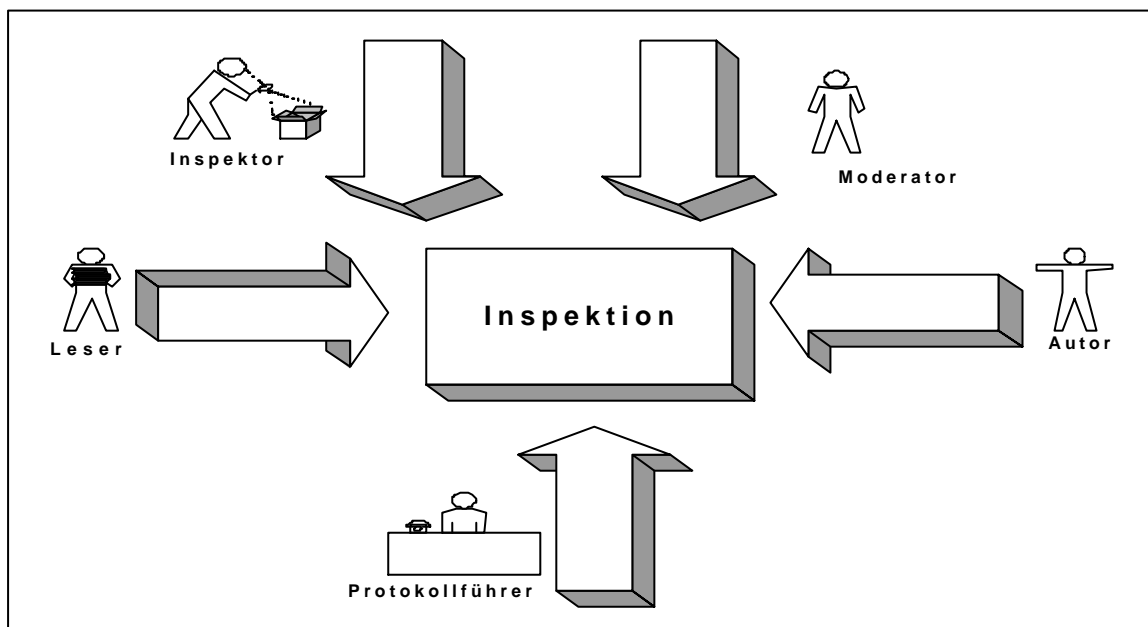


Abbildung5 die Rollen der Inspektion

Der **Organisator** plant alle Inspektionsaktivitäten innerhalb eines Projektes oder zwischen den Projekten.

Der **Moderator** stellt sicher, dass die Inspektionsabläufe richtig abgehandelt werden und dass die Inspektionsteilnehmer ihre Aufgaben richtig erfüllen in jeder Phase des Inspektionsprozesses. Er muss zudem über Moderations- und Leitungsfähigkeit verfügen, da er während der Inspektionssitzung für den gesamten Ablauf verantwortlich ist.

Die **Inspektoren** sind für die Findung aller möglichen Fehler verantwortlich, die sich im zu untersuchenden Artefakt verstecken

können. Inspektor kann jeder im Team sein, unbeachtet ob er schon eine Rolle zugewiesen bekommen hat oder nicht.

Der **Leser** oder Präsentator hat zur Aufgabe die Präsentation des Software Produktes in einem adäquaten Tempo, in dem er die Teammitglieder durch das Produkt vollständig und logisch durchführt. Die Durchführung bedeutet nämlich nicht das Vorlesen, sonder vielmehr die Interpretation und Erklärung der Artefaktsmaterie.

Der **Autor** ist derjenige, der das Produkt erstellt hat und ist damit für die Behebung der Fehler verantwortlich während der Überarbeitungsphase. Der Autor muss dann die Fragen beantworten, die für den Leser nicht verständlich sind, und darf damit nicht gleichzeitig als Moderator oder Protokollführer fungieren.

Der **Protokollführer** ist für die Erfassung aller Fehler, die während der Inspektionssitzung gefunden wurden in einer Fehlerliste.

Die **Rolle** des Kollektors findet man nur bei Inspektionsmethoden, die keine Inspektionstreffen vorsehen. Der Kollektor ist dann für das Sammeln aller Fehler zuständig.

Nun muss man wissen, wie groß das Team sein soll. Über die Teamgröße wird in der Software Kommune gestritten. Manche empfehlen kleine Teams von zwei bis drei Inspektoren, andere dagegen bevorzugen größere Teams von über sechs Personen. Wir beziehen uns hier auf die Empfehlung von Laitenberger [Lai1a], der die Aufstellung eines Teams von drei oder vier Leuten, mit einem Moderator, einem Autor und einem oder zwei Inspektoren vorschlägt. Man könnte dann jeder Zeit neue Inspektoren dazunehmen, wenn man feststellt, dass die Inspektion doch noch mehr Leute erfordert.

Zuletzt möchten wir darauf hinweisen, dass die Inspektoren am besten direkt aus der Entwicklungsumgebung des Produktes sein sollen. Inspektoren sollen generell gute Erfahrung und Wissen, was Inspektionen sind und wie sie ablaufen, mitbringen.

Auch nach der Fertigstellung jeder Softwarekomponente oder Software System können diese gegen die Anforderungen validiert werden. Diese Vorgehensweise nach dem V-Modell erhöht auch in dieser Richtung die Verfolgbarkeit [Rom01].

Das V-Modell bietet eine gute Grundlage für die Inspektionen, aber ist nicht für die Durchführung von Inspektionen erforderlich. Inspektionen lassen sich natürlich auch an anderen Modellen orientieren. Sie sind unabhängige Techniken zum Finden von Fehlern in der Software Entwicklung, und können daher immer und überall angewandt werden.

2.8 Prüfverfahren im Software Engineering

Die Verfahren im Software Engineering reichen von konstruktiven Maßnahmen zur Fehlervermeidung bis hin zu analytischen Methoden zur Fehlerentdeckung.

Unter den analytischen Maßnahmen fallen auch die Lesetechniken und die formalen Inspektionen [Tra96].

Unter Lesetechniken versteht man die individuelle Analyse eines Software Artefaktes, um ein besseres Verständnis für eine Aufgabe zu erlangen. Diese Aufgabe beschränkt sich nicht nur auf die Fehlerfindung, sondern umfasst alle möglichen Aufgaben im Software Engineering, z.B. Wiederverwendung, Wartung, etc...

2.8.1 Informale Lesetechniken

2.8.1.1 Ad-hoc

Die **Ad-hoc** Technik ist die erste Inspektionsmethode überhaupt. Fagan [Faf76] wollte zuerst eine Technik, die ohne großen Aufwand auf Code anzuwenden wäre, um die Fehler herauszufinden. Dabei wird ohne große Anpassung ein Dokument auf Fehler hin untersucht. Die Inspektoren erhalten keinerlei Anleitung oder technischen Hilfe, wie oder worauf das Dokument untersucht werden soll. Deshalb sind hier die Erfahrung und Wissen der Inspektoren von enormer Wichtigkeit. Die Ad-hoc Methode wird heute noch oft praktiziert, da

sie schnell und überall einsetzbar ist. Diese Methode bietet keine Möglichkeit zur Verfolgbarkeit, da sie der subjektiven Meinung der Inspektoren unterliegt.

2.8.1.2 Checklisten-basierte Lesetechnik

Die **Checklisten-basierte Lesetechnik** bietet den Inspektoren mehr Unterstützung. Diese Hilfe ist in Form von geschlossenen Fragen, auf die die Inspektoren vor, während oder nach dem Lesen antworten sollen. Diese Checklisten bilden den Baustein der Untersuchung. Die Fragen sind entweder so allgemein formuliert, oder werden aus gesammelten Erfahrungen und Projekten hergeleitet, dass dies zur Last dieser Untersuchungstechnik angerechnet wird. Außerdem passiert die Beantwortung dieser Fragen rein subjektiv, und bietet Dritten keine Möglichkeit zur Verfolgbarkeit von Fehlern [Tra96].

2.8.1.3 Walkthrough

Walkthroughs werden unter Führung des Autors durchgeführt. Der Autor versucht das Produkt der Gruppe verständlich zu machen, indem er oder sie die Gruppe durch das Dokument hindurch begleitet. Dabei werden interaktive Analysetechniken verwendet. Das Ziel dabei ist, die Problembereiche zu entdecken. Die entdeckten Fehler können allerdings erst nach Abschluss des Walkthroughs, und nur mit Zustimmung des Autors behoben werden. Verantwortlich dafür ist der Walkthroughleiter [FW90].

2.8.2 Formale Inspektionsmethoden

2.8.2.1 Audit

Bei **Audits** geht es um die systematische Überprüfung der Qualitätsmerkmale eines kritischen Softwareproduktes durch qualifizierte unabhängige Dritte, seien es Personen oder Institutionen. Das Ziel dabei ist die Feststellung von Abweichungen zu den Spezifikationsdokumenten, Vertragsbedingungen und zu Entwicklungsstandards. Häufig werden vor Auslieferung eines Software Produktes ein **funktionaler** und ein **physikalischer Audit**

durchgeführt. Dabei werden formale und operationale Techniken verwendet [Tra96].

In dem funktionalen Audit werden die Anforderungen auf Erfüllbarkeit untersucht, während im physikalischen Audit der Quellcode und seine Dokumente auf Vollständigkeit, Konsistenz und Lieferbarkeit untersucht werden.

2.8.2.2 Review

Reviews sind umfassende und systematische Prüfverfahren der Qualitätsmerkmale eines Softwareproduktes. Sie dienen der Überprüfung von Meilensteinen am Ende jeder Entwicklungsphase. Dabei wird ein Produkt oder eine Menge von Produkten den Projektteilnehmern vorgestellt. Die Teilnehmer können Manager, Kunden, Benutzer oder Dritte sein, die während der Reviewsitzung das Produkt kommentieren und auf die möglichen Fehler hinweisen. Je nach Ausgang einer Reviewsitzung, wird entschieden, ob die nächste Phase angegangen werden darf, oder die vorige erst überarbeitet werden muss [FLS91]. Je nach Phase unterscheidet man: Anforderungsreview, vorläufiger Entwurfsreview, kritischer Entwurfsreview, Codereview und Abnahmetestreview.

2.8.2.3 Kontrollflussorientiertes Lesen

Bei dieser Lesetechnik muss der Gutachter einen Kontrollfluss lesen und überprüfen, in dem er den Variablen, Konstanten und ähnliche Parameter im Dokument mit möglichen und sinnvollen Werten belegt, und das Verhalten des Kontrollflusses bei diesen Werten untersucht. Die Untersuchung des Verhaltens eines Kontrollflusses beinhaltet den Vergleich des Soll Verhaltens mit dem Fehlverhalten. Dabei werden die Fehler und Fehlerursachen entdeckt und isoliert. Wie der Name schon sagt, kann diese Lesemethode nur für Dokumente, die einen Kontrollfluss enthalten, in Frage kommen [Lig02].

2.8.2.4 Lesen durch schrittweise Abstraktion

Diese Lesetechnik wird nicht häufig in der Literatur zitiert und beschrieben. Das **Lesen durch schrittweise Abstraktion** (im

Englischen: Reading by Stepwise Abstraction) wurde 1972 von Harlan D. Mills erstmals entwickelt. Im wesentlichen wird diese Technik auf Code angewandt. Jeder Inspektor muss eine Sequenz im Code sich durchlesen, und die elementaren Bausteine eines Programmcodes, wie z.B. Prozeduren oder Berechnungsfunktionen dann zu größeren Bauelementen abstrahieren. Dieser Vorgang wird mehrmals wiederholt bis die letzte Funktion des Codes abstrahiert wurde, und gegen seine Spezifikation verifiziert werden kann. Sowohl bei der eigentlichen Abstraktion als auch beim Vergleich können Fehler entdeckt werden [ROM01].

Da diese Technik nur auf Code begrenzt ist, bietet sie den Inspektoren eine formale Möglichkeit, die funktionale Korrektheit des Codes zu testen.

2.8.2.5 Aktive Design Review

Die **Aktive Design Review** (ADR) ist eine Inspektionsmethode speziell für Designdokumente. Dabei versuchen der Designer, der ja auch der Autor des Designdokumentes ist, und der Inspektor die Defekte zu finden. Der Inspektor wird anhand seiner technischen Fähigkeiten ausgesucht. Er ist für die sorgfältige Abdeckung aller Aspekte der Designdokumente, und somit für die Findung der Fehler und Abweichungen verantwortlich [Lai1a].

Der Prozess des ADR besteht aus drei Teilen. Zuerst präsentiert der Autor, also der Designer eine Übersicht über sein Produkt, und es wird der Review Termin festgesetzt. Im nächsten Schritt wird die ADR-Sitzung durchgeführt. Dafür bereitet der Autor eine Fragenliste vor, um den Inspektor durch das Produkt durchzuführen. Diese Fragen sind so genau auf das Dokument abgestimmt, dass der Inspektor nur mit sorgfältiger Studie des Dokumentes die Fragen beantworten kann. Der letzte Schritt bei dieser Technik ist die Fehlerkollektion. Dazu dienen die Inspektionssitzungen. In jede Sitzung wird eine Qualitätseigenschaft des Artefaktes überprüft.

2.8.2.6 Szenario-basierte Lesetechniken

Unter **Szenario basierten Lesetechniken** verbergen sich vor allen Dingen die unten angeführten Inspektionsmethoden: Perspektiven

basierte-, Defekt basierte- und Function Point basierte Inspektionen. Die Idee dahinter ist, die Inspektoren in bestimmten Szenarien zu versetzen, um spezielle Defekte finden zu können [Lai1a].

Ein Szenario ist eine Beschreibung von Interaktionssequenzen meistens zwischen einem oder mehreren testenden Benutzern und dem zu testenden Artefaktes. Das Szenario besteht aus einer Liste von Fragen und einer detaillierten Einleitung für die Inspektoren, wie das Dokument zu lesen ist, und wie sie die Information in dem Dokument anzusehen haben. Die Inspektion verläuft im Team von mehreren Inspektoren, die verschiedenen Szenarien bearbeiten sollen. Damit ist gesichert, dass mehrere Gesichtspunkte des Dokumentes behandelt werden, und somit mehrere Qualitätsmerkmale überprüft werden. Dadurch wird eine hohe Überdeckung der möglichen Fehlerklassen erreicht und eine geringe Überlappung [Rom01].

Die Effektivität der Szenario basierten Lesetechniken hängt stark von dem Inhalt und Design der zu inspizierenden Software Artefakten ab.

Wir betrachten nun in den folgenden Abschnitten die drei Variationen dieser Lesetechnik.

2.8.2.6.1 Function Point basierte Lesetechniken

Diese Lesetechnik wurde von Cheng und Jeffery entwickelt. Sie definiert Szenarien für die Fehlerfindung in Anforderungsdokumenten und basiert auf den Ansatz der Function Point Analysis(FPA). Diese Analysis definiert ein Software System entsprechend seiner Eingabe, Files, Anfragen und Ausgabe. Diese vier Elemente bilden dann die Szenarien in dieser Lesetechnik. Ein Function Point Szenario besteht aus Fragen, die den Inspektor dazu leiten sollen, auf einen bestimmten Punkt des Anforderungsdokumentes zu fokussieren [Lai1a].

2.8.2.6.2 Defekt basierte Lesetechniken

Bei dieser Lesetechnik geht es darum, die Anforderungsdokumenten auf möglichst viele verschiedene Fehlerklassen hin zu untersuchen. Dazu definiert man verschiedene Fehlerklassen. In jeder Fehlerklasse

wird wiederum eine Menge von Fragen definiert. Die Liste von den Klassenspezifischen Fragen bildet ein Szenario.

Der Inspektor muss dann vor, während und nach dem Lesen eines Dokumentes diese Fragenliste beantworten. Die Fehler eines Szenarios fokussieren auf eine oder mehrere Fehlerklassenart, die in der Entwicklungsumgebung häufig auftreten. Dies hilft dem Inspektor, die Fehler dieser bestimmten Klasse besser zu finden. Jedes Szenario bestimmt die Fehlerklasse, die die Gutachter bestimmen müssen. Aus dieser Idee heraus hat sich die Perspektiven basierte Lesetechnik weiterentwickelt [Lai1a].

2.8.2.6.3 Perspektiven basierte Lesetechniken

Die Idee bei der Perspektiven basierten Lesetechnik ist eigentlich intuitiv. Ein Problem, das wir aus verschiedenen Blickwinkeln betrachten können wir besser und vor allen Dingen ganzheitlich lösen. Denn durch die verschiedenen Sichten werden auch verschiedene Ursachen entdeckt.

Diese Tatsache macht sich das Perspektiven basierte Lesen zu eigen.

Diese Methodik dient der systematischen Überprüfung von Software Dokumenten und basiert auf die Inspektion eines Software Artefaktes aus verschiedenen Perspektiven von unterschiedlichen Interessensvertretern. Diese Perspektiven hängen hauptsächlich davon ab, welche Rolle die Inspektoren innerhalb des Entwicklungsteams haben. Die Software Artefakte werden dann aus jeder sinnvollen Perspektive betrachtet und inspiziert, was die Fehlerfindung erheblich steigert.

Die **Perspektiven** sind operationale Szenarien, die die Beschreibung einer oder mehreren Aktivitäten beschreiben, und eine Menge von Fragen beinhalten. Sie beinhalten eine Anleitung für die Gutachter wie die Dokumente aus der entsprechenden Perspektive zu lesen sind, eine Menge von Aktivitäten und eine dazu entsprechende Menge von Fragen. Dabei können nur Fehler entdeckt werden, die für die vorgegebene Sicht relevant sind.

Dadurch dass die Gutachter in die Lage einer bestimmten Perspektive versetzt werden, wird auch eine gewisse Unabhängigkeit von externen Faktoren, wie die persönliche Erfahrung des einzelnen Gutachters, mit dem selben Projekt oder mit anderen Projekten gewährleistet. Die Ausführung der angegebenen Aktivitäten hat zum Ziel, das Lenken der Gutachteraufmerksamkeit auf bestimmte Aspekte des Dokumentes, wodurch das Verständnis des Dokumentes garantiert wird. Dieses Verständnis bildet wiederum die Grundlage für eine bessere Dokumentenanalyse und somit unterstützt es dem Gutachter beim Finden von Fehlern.

Zusätzlich zu diesen speziellen Fehlern, können die Inspektoren auch allgemeine oder andere Fehler finden, die nicht aus dem Szenario hervorgehen, sondern aus der persönlichen Erfahrung oder aus sonstigen Kenntnissen. Sie werden dann auch dazu angehalten, diese Fehler gesondert zu dokumentieren.

Der Vorteil der PBR-Lesetechnik ist offensichtlich, wenn man daran denkt, dass die meisten Qualitätsmerkmale keine einheitliche Definition haben, z.B. wird Korrektheit aus Sicht des Anwenders anders definiert als die Korrektheit aus Sicht des Entwicklers. Genauso ist es mit den anderen Qualitätseigenschaften, wie Zuverlässigkeit, Testbarkeit, Wartbarkeit und Benutzerfreundlichkeit. Daher stellt die Perspektiven basierte Technik ein oder mehrere Szenarien für jede Perspektive zur Verfügung. Diese Szenarien basieren im wesentlichen auf die Aktivitäten, die die Inspektoren selbst im Entwicklungsteam betreiben, und versetzen sie in die Lage, das Produkt aus der entsprechenden Perspektive noch besser beurteilen zu können. Dadurch bekommt man eine umfassende Artefaktbeurteilung und auch eine hohe Abdeckung aller möglichen Fehlerursachen.

2.9 Beurteilung von Inspektionstechniken

Überlappung und **Überdeckung** sind zwei Kriterien zur Beurteilung von Lesetechniken. Unter Überlappung versteht man, wie viele gemeinsame Fehler von den Gutachtern beim Lesen eines Dokumentes gefunden werden. Mit der Überdeckung ist die Anzahl der beim Lesen eines Dokumentes gefundenen Fehler gemeint.

Wichtig ist festzuhalten, dass eine geringe Überlappung die Wahrscheinlichkeit einer hohen Überdeckung steigert und umgekehrt. Eine gute Lesetechnik zeichnet sich durch eine geringe Überlappung und eine hohe Überdeckung. Folgende Abbildung versucht diesen Sachverhalt anschaulich zu machen.

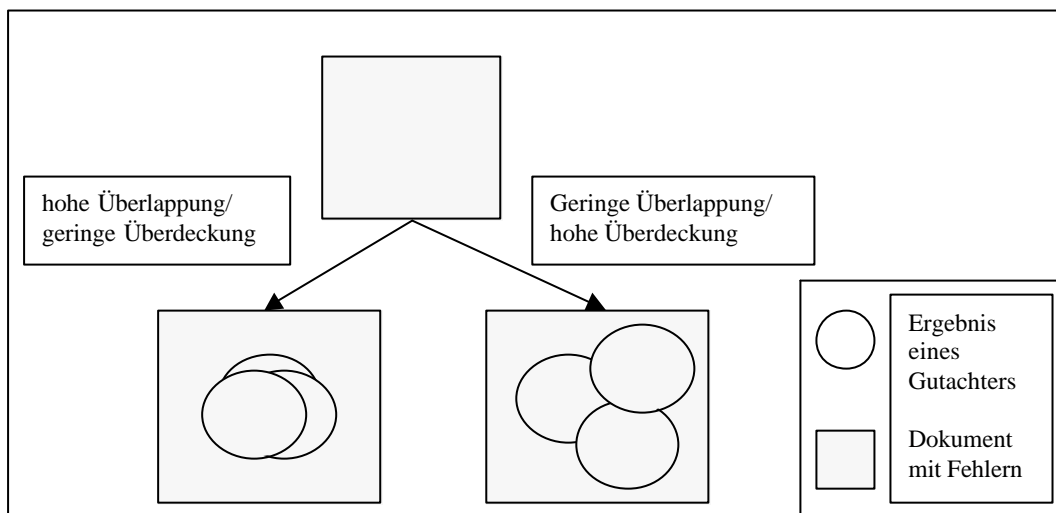


Abbildung 8 Überlappung und Überdeckung innerhalb eines Dokumentes

Die Szenario basierten Lesetechniken im Allgemeinen und die Perspektiven basierten Lesetechnik im speziellen zeichnen sich durch eine hohe Überdeckung und eine geringe Überlappung aus. Wogegen sich die Ad-hoc Methode z.B. sich durch eine geringe Überdeckung und eine hohe Überlappung auszeichnet.

2.10 Modelle der verschiedenen Lesetechniken

Bei den Lesetechniken gibt es zwei grundsätzliche Ansätze, die die Orientierungsgrundlage für die Prüfverfahren bilden. Dies sind der Dokumentorientierte Ansatz, und der Architektur orientierter Ansatz [Lai00].

2.10.1.1 Der dokumentorientierte Ansatz

Dieser Ansatz beruht auf die Tatsache, dass die Inspektion sich mehr an den Dokumenten selbst orientiert, sei es Anforderungsdokumente, Design- oder Quelldokumente, und nicht an ihren Strukturen noch

an dem Informationsgehalt der Dokumente. Dieser Ansatz hat zwei schwerwiegende Nachteile.

Zum einen ist dieser Ansatz nicht skalierbar. Denn solange die Inspektion sich am Dokument orientiert, besteht die Möglichkeit, dass der Inhalt des Dokumentes zu umfangreich wird. Deshalb müssen die Dokumente in Teildokumente aufgeteilt werden und die dazugehörigen Inspektionen in mehreren Sitzungen durchgeführt werden. Dies allerdings kann von Nachteil werden, wenn der logische Zusammenhang an dieser Aufteilung leidet.

Zum anderen sind die wichtigen Informationen über das ganze Dokument verteilt, oder sogar über verschiedene Dokumenttypen. Das letztere ist besonders der Fall, bei der Objekt orientierten Software Entwicklung und dem Einsatz von Modellierungssprachen wie die Unified Modelling Language (UML). Es führt zu großen Schwierigkeiten, wenn die Inspektion nur für Teildokumente vorgesehen wird, und somit wichtige Teildokumente, die zu dem gesamten Verständnis des Software Artefaktes, gar nicht bei der Inspektion in Betracht gezogen werden können.

Die Ad-hoc Methode und die Checklisten-basierte Lesetechniken verfolgen wie die meisten Prüfverfahren den Dokument orientierten Ansatz [Lai00].

2.10.1.2 Der architekturorientierte Ansatz

Die Lösung zu den oben angedeuteten Problemen ist der Architektur orientierter Ansatz. Dieser Ansatz unterscheidet zwischen der logischen Einheit eines Software Systems und ihrer Dokumentation. Zu jeder logischen Einheit existiert ein Dokument. Ein Dokument kann die relevanten Informationen zu einer oder mehreren logischen Einheiten enthalten. So bekommen wir eine klare Unterscheidung zwischen den logischen Einheiten und dazugehörigen Dokumenten. Eine logische Einheit kann beispielsweise ein Konzept oder eine Einheit im Dokument sein, dagegen könnte ein Dokument z.B. eine Beschreibung sein. Allgemein beschreibt Laitenberger mit logischer Entität, alles was die unsichtbare und konzeptuelle Eigenschaft einer

Software übermitteln kann, und mit Softwaredokumentation alles greifbare, das Gegenstand einer Inspektion sein kann [Lai00].

Abbildung 8 gibt uns eine mögliche Darstellung davon.

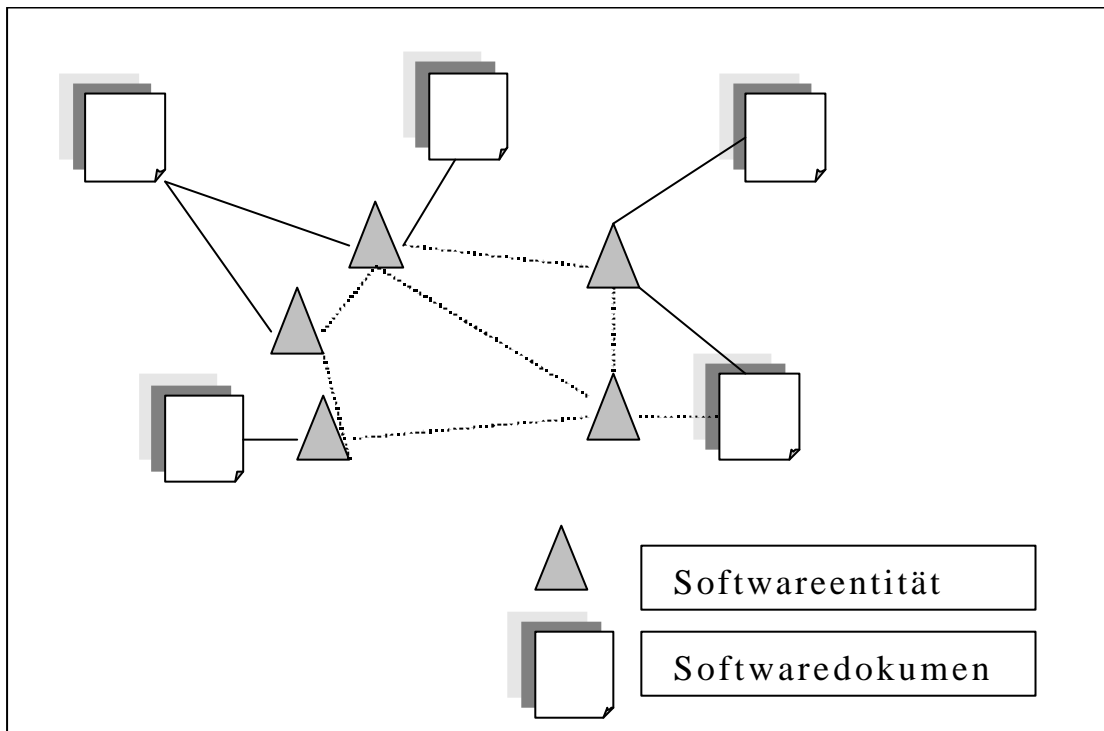


Abbildung 9 Logische Entitäten und deren Dokumentationen

Die Vorteile dieses Ansatzes sind wie folgt:

Jede logische Einheit wird individuell definiert und enthält daher alle für sie relevanten und in ihr enthaltenen Informationen. Daher vermittelt das dazugehörige Dokument alle wichtigen Informationen, die für eine Inspektion notwendig und ausreichend sind.

Dieser Ansatz ist skalierbar. Dies bedeutet, dass jede logische Einheit selbst in logischen Einheiten unterteilt ist, sodass immer gesichert ist, in einer Inspektionssitzung, die schaffbare Menge an Informationen überprüft werden kann

Dieser Ansatz verfolgt intuitiv den objekt-orientierten Ansatz, aber kann in allen anderen Software Entwicklungsansätzen auch problemlos übertragen werden.

Die Perspektiven basierte Inspektion verfolgt diesen Ansatz.

2.11 Zusammenfassung

In diesem Kapitel haben wir uns mit den Inspektionsmethoden beschäftigt. Denn ohne Inspektionstechniken können keine qualitativen Aussagen über Softwaresysteme gemacht werden.

Software Inspektionen dienen dazu, die während des Entwicklungsprozesses eines Software Systems generierten Dokumente zu analysieren. Deshalb zählen sie zu den analytischen Strategien. Zu den Dokumenten gehören unter anderem die Produktanforderungen, Entwürfe und der Quellcode. Software Inspektionen werden auch statische Verifikationsmethoden genannt, weil das zu entwickelnde System dazu nicht ausgeführt werden muss.

Diese Inspektionstechniken gliedern sich in statische und dynamische Methoden. Die Unterteilung geht soweit bis hin zu informellen und formalen Methoden wie die folgende Abbildung erläutert [Lig02]:

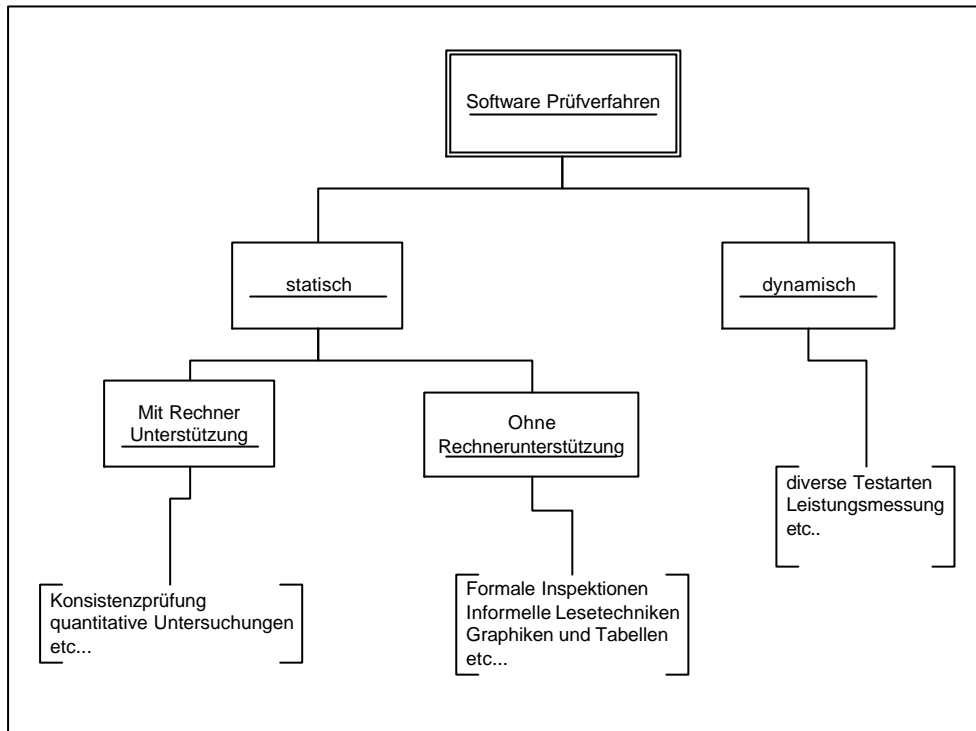


Abbildung 10 Lesetechniken

Wir haben diese Prüfverfahren und deren Klassifikation im Einzelnen definiert. Zuvor haben wir den Prozess der Inspektion selbst kennengelernt und seine Aufteilung in Phasen definiert. Die Verfeinerung dieser Aufteilung hängt von der ausgewählten Technik ab, manche Phasen sind auch optional und können aus Kostengründen nach reiflicher Überlegung auch weggelassen werden. Ein Inspektionsprozess sieht Rollen für die Inspektionsmitglieder vor. Diese sind: der Moderator, der Leser, der Autor, der Protokollführer und die Inspektoren.

Um das Kapitel abzurunden, haben wir noch die beiden Ansätze eingeführt, die eine Lesetechnik verfolgen kann. Es sind der dokumentorientierter Ansatz und der architekturorientierter Ansatz.

Die Leistungsfähigkeit der einzelnen Prüfverfahren hängt von der Gründlichkeit der Inspektionsmethoden ab und von der Aufteilung des Inspektionsprozesses ab, wie zeitaufwendig jede Phase ist. Diese Gründlichkeit definiert zum einen die Effektivität einer Lesetechnik,

und zum anderen wirkt sie sich auf den gesamten Aufwand und Kosten der Inspektion.

Zu diesem Thema sind zahlreiche Veröffentlichungen erschienen. Hier sei auf [Lai1a], [Tha96], [Fag76], [Fag86] und [Rus91] hingewiesen.

Dieses Kapitel bildet eine wichtige Grundlage für diese Diplomarbeit, da bei der Entwicklung von Wissensinspektionen Bezug auf die Softwareinspektionen genommen werden soll. Daher musste das Thema Softwareinspektionen intensiv behandelt werden.

Das nächste Kapitel befasst sich mit den Themen Wissen und Wissensmanagement. Dabei sollen Wissensmerkmale und -problembereiche identifiziert werden, bevor entsprechende Methoden für die Inspektion von Wissen im übernächsten Kapitel entwickelt werden können.

3 Von Wissen zu Wissensmanagement

Dieses Kapitel behandelt Wissen und sein Management. Es wird versucht zu definieren, wo und wann genau Wissen anfängt. Der Begriff „Wissen“ ist vielseitig zu verstehen und hat auch tatsächlich verschiedene Definitionen. Ich werde hier diese Definitionen auf vielen Ebenen vorstellen. Es wird auch der Prozess der Wissensakquisition vorgestellt, bevor die Frage nach den Wissensträgern gestellt und beantwortet.

Wissen ist nicht nur in seine Definitionen vielfältig, sondern auch in seinen Darstellungsformen. Deswegen werden diese auch Gegenstand dieses Kapitels.

Bei einer Diplomarbeit im Bereich Informatik, die u.a. von Wissen handelt, stellt sich die Frage nach dem Wissensmanagementkonzept. Daher wird dieses Thema hier angegangen und die Wissensbausteine definiert. Anschließend werden die verschiedenen Typen von Wissensmanagement Systeme vorgestellt.

3.1 Wissen

Bevor wir Konzepte zur Wissensinspektion entwickelt werden können, muss erst der Begriff Wissen definiert werden. Ist Wissen klar und präzise, oder ist Wissen etwas allgemeines, mehrschichtiges und aus den verschiedensten Komponenten zusammensetzbar?

- Wenn man nach Wissen in Wahrig guckt, findet man folgendes: **'Wissen** <n.; -s; unz.> Kenntnisse, Gelehrsamkeit; Kenntnis, Bewusstsein (von etwas) [Wah91];
- Im Brockhaus steht: **'Wissen** ist die begründete und begründbare Erkenntnis im Unterschied zur Vermutung und Meinung oder zum Glauben'

Ist dann Wissen einfach die Gesamtheit der Kenntnisse? Was hat dann noch das Bewusstsein damit zu tun? Spielen noch andere Faktoren eigentlich noch eine Rolle?

- Interessant ist die Definition von Sveiby [Sve99]:
„ **Wissen** kann bedeuten: Information, Erkenntnis, das Kennen, Kenntnis, Wahrnehmung, Weisheit, Wissenschaft, Erfahrung, Fertigkeiten, Einsicht, Kompetenz, Erfahrungswissen, praktische Fähigkeiten, Intelligenz, Lernfähigkeit, Weisheit, Gewissheit, usw. “
- Die zweite interessante Definition von Wissen, ist die von Davenport und Porsak [DP98]. Die am besten die messbaren und unmessbaren Charakteristiken von Wissen darlegt:
„ **Wissen** ist eine flüssige Mischung eingerahmter Erfahrung, Werte, dem Zusammenhang gemäß Informationen, fachmännischen Einblicks und geerdeter Intuition, die eine Umgebung und ein Gerüst für die Bewertung und Vereinigung von neuen Erfahrungen und Informationen versorgt. Es kommt aus den Empfindungen von Wissenden und wird auch dort wieder angewendet. In Organisationen wird es oft nicht nur in Dokumenten oder Quellen eingebettet, sondern auch in organisatorischen Routinen, Verfahren, Praxis, und Normen.“
- Probst [Pro97] fasst vieles zusammen und sagt zum Wissen:
„ **Wissen** bezeichnet die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen.“

Aus diesen Definitionen wird klar, dass Wissen auf jeden Fall vielmehr als nur die Kenntnis oder die Erkenntnis. Vielmehr setzt sich Wissen aus mehreren Komponenten zusammen, und ist auch sehr vom gegebenen Kontext abhängig. So kann sich die Erfahrung auf das Gelernte positiv auswirken und die Kompetenzen des Individuums letztendlich verbessern.

Der Zusammenhang der verschiedenen Komponenten prägt die Schwierigkeit der Definition des Begriffes Wissen aus. So spielen in

der Soziologie andere Faktoren eine Rolle zur Definition von Wissen als in den Naturwissenschaften.

In der Soziologie wird zum Beispiel Wert auf das Bewusstsein und Wahrnehmung Wert gelegt. Während in den Naturwissenschaften mehr auf die Information und Erkenntnissen, die aus fundierten Versuchen gewonnen werden, vertraut wird. Für diese Diplomarbeit ist schon mal hier festzuhalten, dass diese aus Sicht der Naturwissenschaften wichtigen Merkmale von Wissen von großer Relevanz sind.

Wissen unterliegt aber auch noch dem betrachteten Kontext, und wird auch zudem subjektiv interpretiert, beurteilt und angewendet. Daher kommt die Frage auf, kann man Wissen trotzdem objektiv und wahrheitsgemäß darstellen?

Das bringt uns zu folgenden Fragen:

- Wie kann man Wissen überhaupt darstellen?
- Gibt es eine Art von Wissen oder eher verschiedene Arten?
- Ist Wissen die Information, die daraus zu gewinnen ist oder macht die wahrheitstreue Information das Wissen aus?

Um diese Fragen zu beantworten, müssen erst mal die Unterschiede und die Grenzübergängen zwischen Wissen und Information definiert werden.

Vorher lassen Sie uns folgendes festhalten: In dieser Diplomarbeit wird unter Wissen dem zu untersuchenden Objekt verstanden.

3.2 Der Anreicherungsprozess: Daten zu Information zu Wissen

In der Informatik werden Daten aller Arten und Formen bearbeitet. Dies wird mit Datenverarbeitung beschrieben. Im Laufe der Zeit wurden die Daten so groß, dass nur noch die Rede von Informationsverarbeitung, Informationsmanagement und Informationssysteme ist. Leider wird Wissen selbst häufig mit Information verwechselt. Allerdings ist Wissen vielmehr als nur

Information, wie wir sehen werden. Deswegen geht nun der Trend verstärkt in Richtung Wissensmanagement und Wissensbasierte Systeme.

Festzuhalten ist allerdings, dass es eine Beziehung bzw. eine deutliche Abgrenzung zwischen Information und Wissen besteht. Diese Zusammenhänge werden mit dem Anreicherungsprozess in der Literatur beschrieben. Dieser Prozess beschreibt die Beziehung zwischen diesen drei Begriffen.

Häufig wird in der Literatur das unten stehende Bild benutzt, um die Abgrenzungen zwischen den einzelnen Begriffen zu verdeutlichen. Noch gründlicher ist die Begriffshierarchie von Probst, die ausgehend von den Zeichen als syntaktische Basis für die Daten ausgeht, Vergleiche [Pro98].

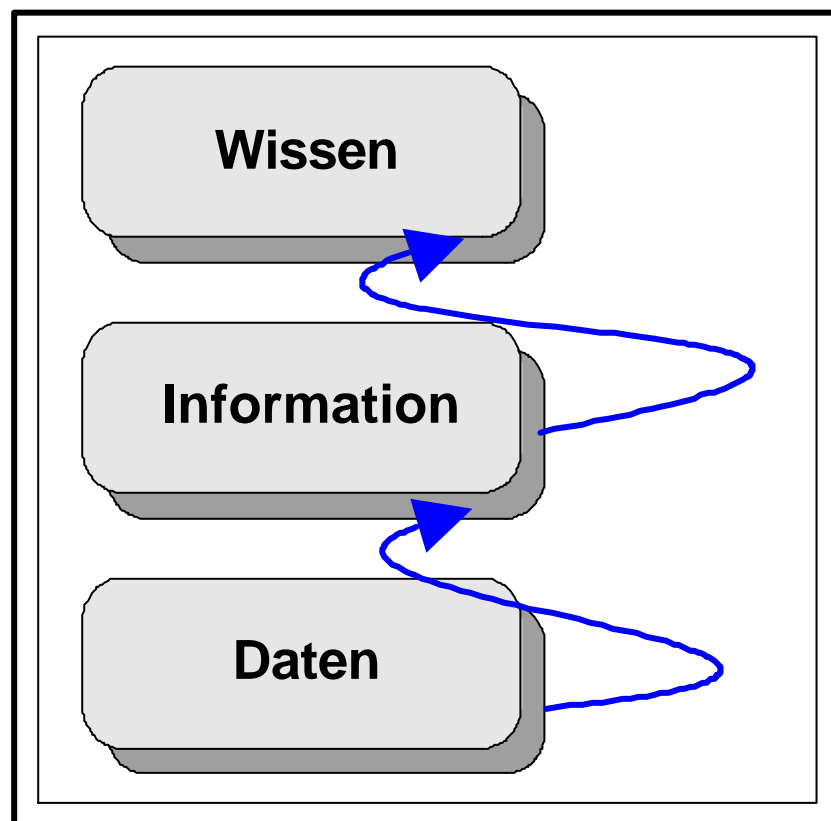


Abbildung 11

Beziehung zwischen Daten, Information und Wissen

In diesem Bild geht man von den Daten als unterste Ebene aus. Aus Daten werden Informationen, wenn diese in einem bestimmten Kontext interpretiert werden können. Auf dieser Weise stellen Daten für den Empfänger Informationen dar. Die Interpretation heißt nichts anderes als die Abbildung von diesen Daten auf ein konkretes Problem oder eine bestimmte Situation. Der Empfänger kann nun Mithilfe der Interpretation von Daten Information gewinnen. Die verschiedenen Informationen können nun wieder miteinander vernetzt werden. Dann können sie in einem konkreten Handlungsfeld genutzt werden. Dies wird nun mit Wissen beschrieben.

Die Nutzung von Daten und Informationen in der richtigen Form zum richtigen Zeitpunkt ist eins der wichtigen Elemente von Wissensmanagement. In diesem Zusammenhang wird der Anreicherungsprozess von Willke um die Begriffe Intelligenz und Reflexion erweitert [Wil96]. Willke meint, dass wenn ein System eine Beziehung zwischen dem vorhandenen Wissen und seinen Zielen herstellt, und ausgehend von diesem Wissen dann handelt, von Intelligenz gesprochen werden kann. Die Reflexion stellt nach Willke eine Kontrollebene für das Wissen dar, um Wissen und Wissenswirkungen zu untersuchen und daraus Wirkungszusammenhänge abzuleiten [Wil96].

Wissen stützt sich auf Daten und Informationen, im Gegensatz zu diesen ist es jedoch immer an Personen gebunden. Daher müssen Daten-, Informations- und Wissensmanagement stets zusammenspielen [Pro97].

Das unten stehende Bild hält dieses Faktum für uns noch mal fest.

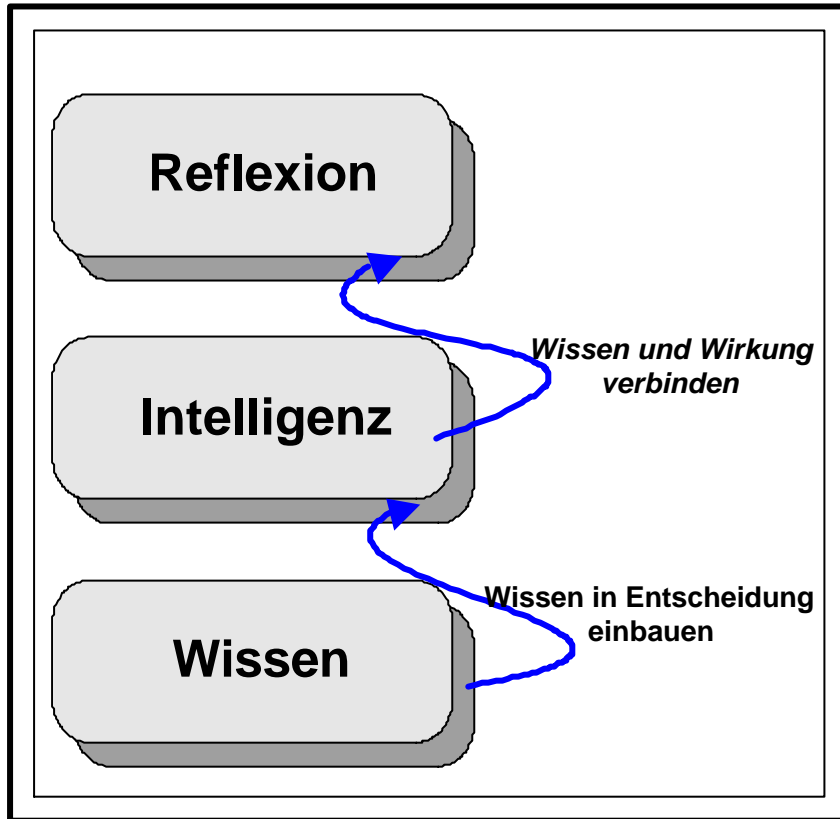


Abbildung 12 Beziehung zwischen Wissen, Intelligenz und Reflexion

Diese Begriffserweiterung ist für diese Diplomarbeit gerade wichtig, da Wissensziele zum Inspizieren unentbehrlich sind. Außerdem haben sie einen Rundungseffekt auf die Bausteine des Wissensmanagements, die wir später kennen lernen werden.

Bevor wir aber die Bausteine des Wissensmanagements angegangen werden können, werden erst mal die verschiedenen Ausprägungen und Arten von Wissen vorgestellt.

3.3 Wissen in der Epistemologie

Die Epistemologie kommt aus dem Griechischen und bedeutet „Wissenschaft des Wissens“. Episteme d.h. wissen und logos d.h. Wissenschaft. Sie ist der Zweig der Philosophie, der sich mit der Frage beschäftigt, wie Wissen, Erkenntnis und Wahrheit prinzipiell zu

erlangen und zu nutzen sind und welche natürlichen Grenzen der Erkenntnis gesetzt sind.

Auf der epistemologischen Ebene wird Wissen in implizites, explizites, latentes Wissen und in Metawissen unterteilt [WN86].

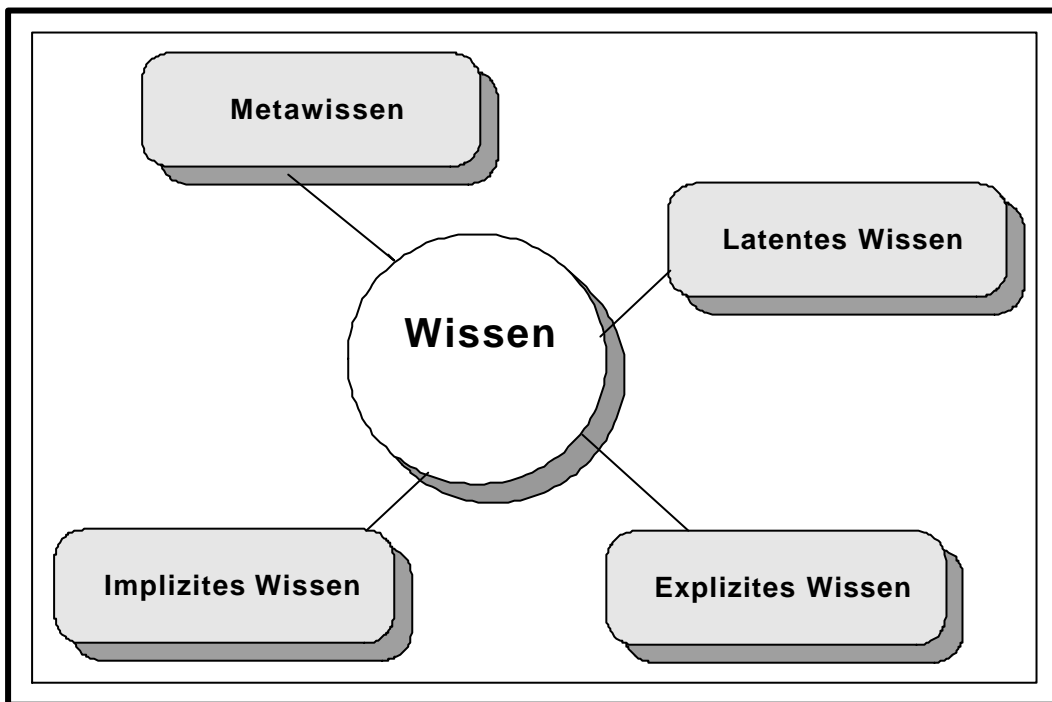


Abbildung 13 epistemologische Wissensformen

- Im **Metawissen** wird Wissen über bestimmte Objekte zur Verfügung gestellt. Darin sind Aussagen über die Zuverlässigkeit und Qualität enthalten sein, sowie Sachverhalte, Ursprung und Umfang des Wissens.
- Das **latente** Wissen ist leichter zu begreifen, da es genau der Gegensatz zum bewussten Wissen ist. Es ist jenes Wissen, das in den Köpfen der Träger hervorschwebt, ohne dass der Träger sich dessen Existenz bewusst ist.
- Das **implizite Wissen** im Englischen „embodied Knowledge“ wird dem Träger(Person oder System) und dessen Erfahrungen, Fähigkeiten und Kompetenzen zugeteilt. Es ist subjektiv und schwer zu formulieren, zu beschreiben oder weiterzugeben.

Es drückt sich als Kompetenzen, Fähigkeiten und Fertigkeiten aus, die durch eine langjährige Erfahrung sich unbewusst akkumulieren. Implizites Wissen ist aber alle mentale Modelle, Glaubens- und Rechtfertigungsmodelle, die das Realitätsbild bestimmen und sich in Idealen Überzeugungen und Gefühle sichtbar macht. Das Letztere wird allgemein mit „tacit knowledge“ zu Deutsch „verborgenes Wissen“ beschrieben. Diese Art von Wissen beschreibt den Bereich, in dem mehr Wissen vorhanden ist als in formaler Sprache ausdrückbar ist. Implizites Wissen ist an Besitzern zeitlich und sozial gebunden und daher wenn überhaupt nur begrenzt verfügbar.

- **Explizites Wissen** „disembodied knowledge“ dagegen berührt den Bereich des greifbaren Wissens. Es hat einen objektiven Charakter und ist jenes Wissen, das standardisiert, methodisch und systematisch in Systemen, Strukturen, Prozessen, Technologien, Dokumentationen, Bibliotheken, Datenbanken und Patenten angelegt ist. Es kann entweder in formaler Sprache oder auch in mathematischen Ausdrücken ausgedrückt werden. Diese Tatsache macht es möglich, dass explizites Wissen von Trägern entkoppelbar ist und separat verwaltet werden kann. Daher ist explizites Wissen allgemein verfügbar und zeitlich stabil.

3.4 Das HANSE-Modell

In der Realität tritt aber Wissen selten in seinen Extremfällen, nämlich rein als implizites oder explizites Wissen auf. Vielmehr ist Wissen als eine Mischung von beiden vorzufinden. Diese beiden Zustände des Wissens haben Wissenschaftler in der ganzen Welt beschäftigt. Einer davon ist David Snowden, der ein Modell entwickelt hat, das diese beiden Wissensformen nach ihren Forderungen deutlicher darstellt [Sno00]. Dieses Modell baut auf den fünf Komponenten: **H**euristiken, **A**rtefakte, **N**atürliche Begabung, **S**kills und **E**r Erfahrungen. Das Modell heißt HANSE(im Englischen ASHEN) Modell. Der Anteil dieser fünf Komponenten hängt stark von dem Bedarf im Prozess, in dem das Wissen benutzt wird [Sch00].

Folgende Abbildung macht den Wissenszustand deutlich zwischen den Extremfällen Implizit und Explizit laut dem HANSE-Modell deutlich:

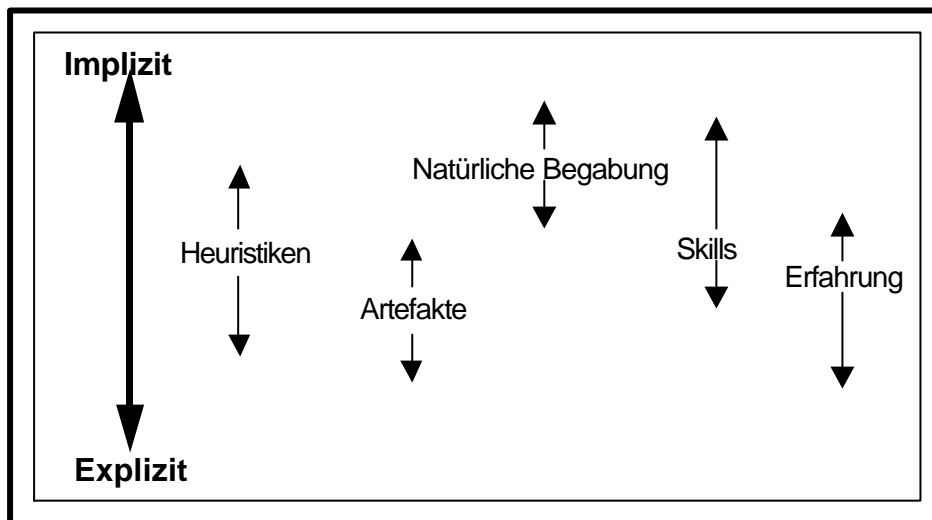


Abbildung 14 Das HANSE-Modell

- **Heuristiken**: Heuristiken sind Daumenregeln, die Fachexperten immer anwenden, wenn ein Rest Unsicherheit bleibt. Sie sind nicht völlig präzise und folgen der Regel, dass 80% auch völlig ausreichen. Es kann sein, dass diese Regeln im Laufe der Zeit explizit werden und doch zu Artefakten anwachsen.
- **Artefakte**: Diese sind alle Bestandteile des expliziten Wissens, wie Datenbanken, Sammlung von Akten oder auch Notizhefte.
- **Natürliche Begabung**: wie der Name hier schon sagt, handelt es sich dabei um die individuelle Genialität eines Wissensträgers, die er in seiner Arbeit einsetzen kann.
- **Skills**: Dieser Begriff ist Englisch und lässt nicht präzise genug im Deutschen wiederfinden. Es beschreibt die Fähigkeiten und Fertigkeiten eines Individuums. Diese Skills sind durch Ausbildung und Übung zu erlangen. Dabei ist für das Wissensmanagement jener Anteil an Skills von Relevanz, dessen Umsetzung gemessen werden kann.

- Erfahrung: Die Erfahrung lässt sich unter stillem implizitem Wissen einzuordnen. Sie lässt sich schwer ausformulieren und beschreiben. Sie wird entweder im Team durchgemacht oder individuell. Die Ergebnisse dabei sind immer situationsabhängig und lassen sich daher nicht verallgemeinern.

Das HANSE-Modell bietet Strategien für die Messung und Umsetzung von implizitem und explizitem Wissen zueinander. Weiterführende Literatur empfiehlt sich das Buch von dem Entwickler Snowden [Sno00] oder das von Schütt [Sch00].

Implizites und explizites Wissen stellen wichtige Ressourcen für das Wissensmanagement dar. Deswegen taucht hier die Frage auf, kann man implizites Wissen zu explizitem machen? Dies wird unter Wissensakquisition beschrieben und im nächsten Abschnitt behandelt.

3.5 Wissensakquisition

Die Transformation von implizitem Wissen zum Expliziten und vom expliziten Wissen zum impliziten Wissen basiert auf das Spiralenmodell von Nonaka [Non97]. Dieses Modell sieht vier Konvertierungsmöglichkeiten: Sozialisation, Internalisierung, Externalisierung und Kombination.

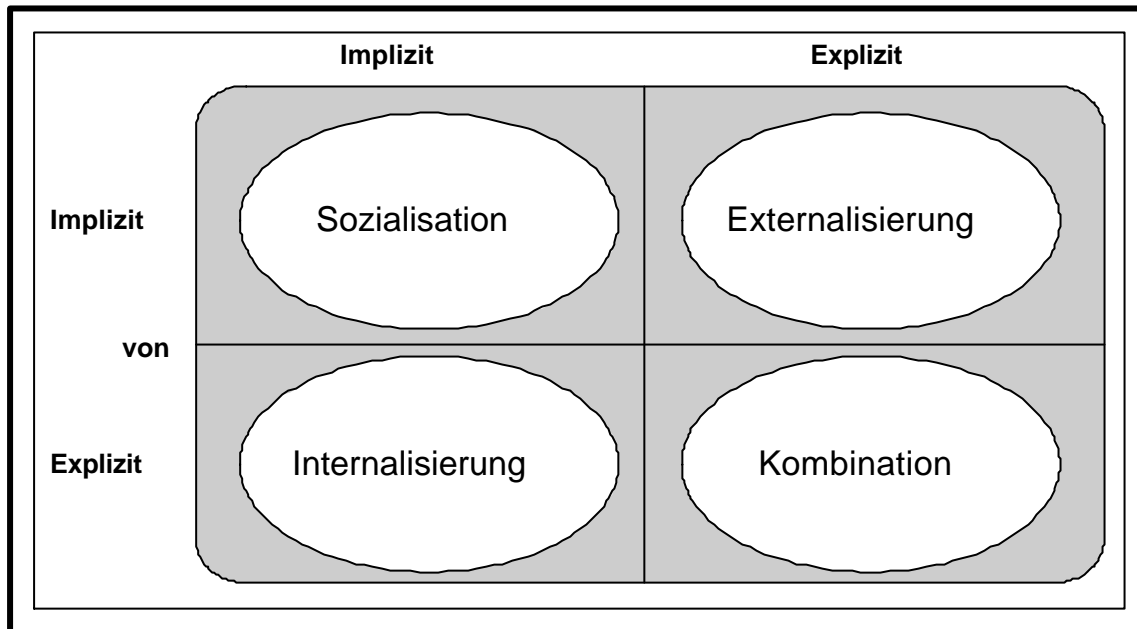


Abbildung 15 Die Wissensspirale

- Die **Sozialisation** beschreibt die Verbindung zwischen implizitem und implizitem Wissen. D.h. dass Wissen unbewusst ohne Explizierung weitergeleitet wird.
- Der Prozess der **Externalisierung** beschreibt den Übergang von dem impliziten Wissen zum expliziten. Wenn dieses explizite Wissen mit einem anderen expliziten Wissen verbunden wird, spricht Nonaka von Kombination.
- Der Prozess der **Kombination** schafft in der Regel immer neues Wissen.
- Der Kreis schließt sich bei dem letzten Prozess der **Internalisierung**, bei dem das explizite Wissen von lernenden Subjekten aufgenommen wird.

3.6 Wissensträger

Wenn wir über Wissen reden, taucht automatisch die Frage nach seinen Trägern auf. Sind es Personen, Dokumente, oder wird heute zutage nur noch digital gespeichert?

Aus verschiedenen Quellen, wie [Wil96], [Wil98], [Pro97], [Pro98] und [FG02].

Fangen wir bei den Personen an. Eine Person erwirbt im Laufe ihres Lebens durch ihre Ausbildung und Erfahrungen eine beachtende Wissensmenge und Fähigkeiten. Dieses Können, Fertigkeiten und Verhaltenweise eines Individuums wird unter **privates Wissen** angeordnet. Somit haben wir die erste Kategorie der Wissensträger: Das private Wissen.

Die zweite Kategorie ist das **kollektive Wissen**. Man spricht von kollektivem Wissen, wenn das Wissen eines gesamten Teams größer ist als das Wissen der einzelnen Teammitglieder. Dieses kollektive Wissen entsteht durch Vernetzung und Verknüpfungen der einzelnen Wissens Elemente.

Dann gibt es noch eine dritte Kategorie von Wissensträgern. Diese umfasst jenes Wissen, das nicht in den Köpfen von Personen gespeichert ist, sondern in den Vorgehensformen eines sozialen Systems wie z.B.:

- Leitlinien einer Organisation
- Standardverfahren
- Arbeitsprozessbeschreibungen
- Routinen
- Merkmale der eigenen Organisationskultur, etc...

Folgende Abbildung fasst für uns die Wissensträger noch mal zusammen:



Abbildung 16 Die Wissensträger

Fazit: für diese Diplomarbeit ist das organisationale Wissen von Wichtigkeit. Es stellt jenes Wissen dar, das inspiziert werden kann, in Form von z.B. Prozess- oder Projektbeschreibungen, interne Dokumente.

3.7 Wissensdarstellungsformen

Nun wissen wir was Wissen ist, welche Wissensformen es gibt und wer seine Träger sein könnten. Es bleibt dennoch die Frage, wie Wissen dann modelliert wird. D.h. wie wird Wissen dargestellt?

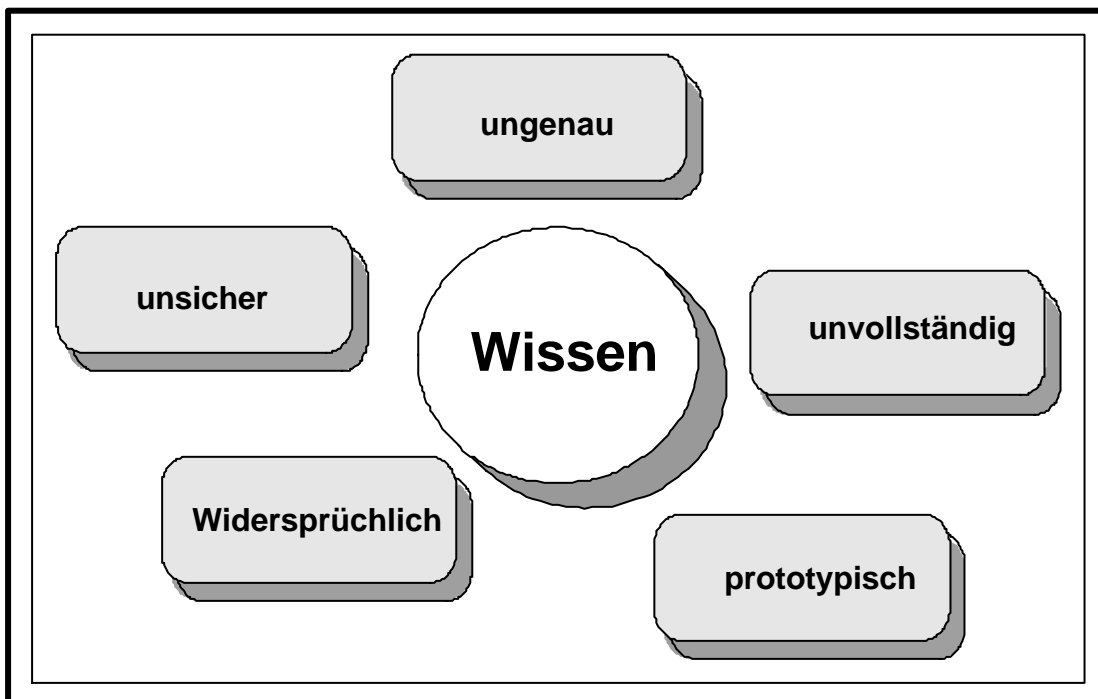


Abbildung 17 Die Wissensmodellierungsformen

Wenn Wissen dargestellt oder modelliert wird, geschieht das in der Regel so, dass bestimmte Aspekte, die einen Bereich vom gesamten Wissen betreffen, abgebildet werden. Dies führt zu einer unvollständigen Darstellung von Wissen. Deshalb wird Wissen in der Literatur in fünf Kategorien klassifiziert:

- Von unvollständiger Darstellung spricht man auch, wenn nur einzelne Darstellungen eines Objektes existieren, z.B. die genaue Beschreibung, aber kein Bild.
- Unsicheres Wissen liegt vor, wenn der Sachverhalt und seine Eigenschaften der Modellierungsperson nicht tiefgründig klar sind.
- Man spricht von ungenauem Wissen, wenn Angaben zu Objektmerkmalen gemacht werden, aber nur in einer ungenauen Form. D.h. nur durch Attribute, oder durch eine Abgrenzung der möglichen Werte. Aber ungenaues Wissen liefert auf jeden Fall mehr Informationen als unvollständiges Wissen, da zumindest Eigenschaftsangaben gemacht werden und nicht komplett fehlen.
- Von prototypischem Wissen spricht man, wenn Wissen nur bestimmte Objektmerkmale beschreibt, die nur für bestimmte Objekte zutreffen und nicht für alle. Diese Art von Wissen schließt nicht aus, dass manche Aussagen auch falsch sein könnten. Allerdings beschreibt es in der Regel nur wahre Aussagen.
- Wenn wir über widersprüchliches Wissen sprechen, dann ist damit die widersprüchliche Modellierung von Wissen. Denn wie wir schon bei der Definition von Wissen gesehen haben, muss Wissen aus der Wahrheit abgeleitet werden und zur Wahrheit führen. Eine solche widersprüchliche Modellierung tritt auf, wenn z.B. Wissen zu unterschiedlichen Zeitpunkten modelliert wird, wobei sich das Wissen in der Zwischenzeit stark verändert hat.

Diese Probleme bei der Wissensdarstellung sind von großer Wichtigkeit bei der Konzeption und Entwicklung von geeigneten

Wissensinspektionsmethoden. Denn es können gute Ergebnisse nur unter Betrachtung der Wissensspezifische Probleme erzielt werden. Im nächsten Kapitel werden wir uns mit diesen Problemen auseinandersetzen.

Aus diesen Erkenntnissen lassen sich schon aber Qualitätsmerkmale für unsere Wissensdokumente ableiten. Ein Beispiel hierzu ist die Forderung, dass Wissen klar und verständlich sein soll. Die kompletten Qualitätsmerkmale werden dann im nächsten Kapitel kennen lernen.

3.8 Wissensmanagement

Seit ein paar Jahren nun ist Wissensmanagement in allen Munden. Es wird so propagiert angesichts heutiger Dynamik als der Kernschlüssel jeden Erfolges. Unternehmen aller Art versuchen nun Mithilfe dieser Strategie Gewinn am Markt zu erzielen, wettbewerbsfähig zu bleiben und sogar dadurch einen Vorsprung zu schaffen.

Aber was steckt eigentlich dahinter? Kann man jedes Wissen managen? Und wie wird das denn gemacht?

Es gibt verschiedene Definitionen von Wissensmanagement. So viele wie Wissensmanagement Experte selbst. Man findet nie eine einheitliche und allgemeingültige Definition. Nichts desto Trotz, lässt sich aus der Recherche sagen, dass sich alle Experten bei folgenden Aussagen einig sind:

- Wissensmanagement hat nichts mit dem Managen von Wissen als solches zu tun, sondern es dient vielmehr der Informationsteilung, Wissensweitergabe, Bildung von Informationssystemen und organisationales Lernen.
- Wissensmanagement beschäftigt sich mit der Analyse einer Organisation, deren Wissenskapital und -kultur, Es versucht das geistige Kapital einer Organisation zu identifizieren und zu strukturieren, so dass neues Wissen entstehen kann, vorhandenes Wissen allen Mitarbeitern zur Verfügung

gestellt werden kann und Informationssysteme innovativ gestaltet werden können.

- Als Ansporn dient die Kernidee, Mitarbeiter einer Organisation dazu zu ermutigen, Informationen aufzubringen, zu teilen und die kollektiven Kenntnisse so anzuwenden, dass sie die optimale Entscheidung treffen können, in Echtzeit.

Aus diesen zusammengefassten Definitionen, können wir drei Hauptsäule des Wissensmanagements erkennen: Der Mensch, der das Wissen schafft und benutzt, die Organisation, die die eigenen prozess- und strukturorientierten Aspekte selbst definiert und die Technologie, die die nötigen Informations- und Kommunikationstechniken zur Verfügung stellt. Diese Sichtweise betrachtet das Wissensmanagement aus einer ganzheitlichen Perspektive und ist an Reinmann-Rothmeier/Mandl angelehnt [Rei98]. Folgende Abbildung hält das noch mal für uns fest:

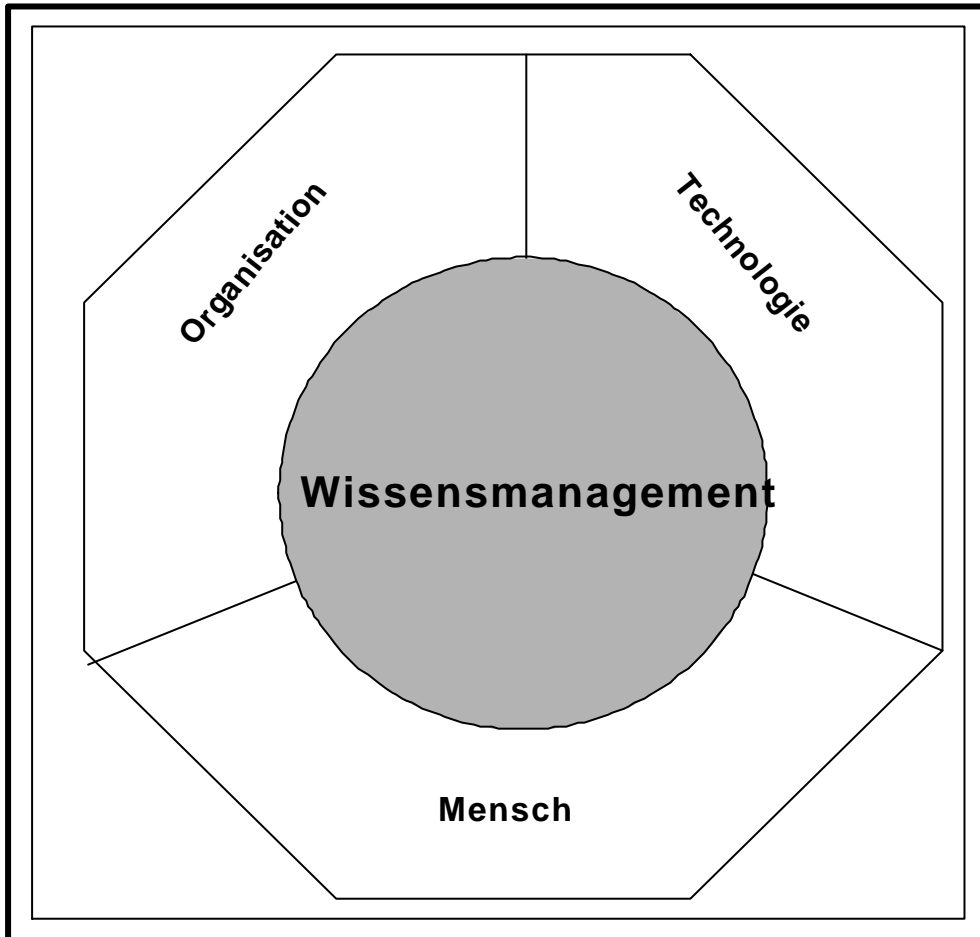


Abbildung 18 Die drei Dimensionen des ganzheitlichen Wissensmanagements

Probst hat aus diesem Ansatz heraus Forderungen an ein pragmatisches Wissensmanagement Konzept entwickelt [Pro97]. Um dies zu erreichen, muss das Konzept folgendes leisten:

- Unternehmensprobleme in Wissensprobleme übersetzen und Entscheidungen in ihrer Wirkung auf organisationale Wissensbestände beurteilen können,
- Pauschallösungen vermeiden und beim Verständnis wissensspezifischer Probleme helfen,
- Sich stets an konkreten Problemen orientieren und die Bodenhaftung nicht verlieren,

- Ein handlungsorientiertes Analyseraster sein und erprobte Instrumente zur Verfügung stellen,
- Kriterien für die Messbarkeit des Erfolges entwickeln,
- an existierende Systeme anschließen und bestehende Lösungsansätze integrieren
- in einer verständlichen Sprache formuliert sein, welche im Unternehmensalltag vermittelbar ist.

Manche dieser Aspekte sind gerade für die Wissens-Inspektion von großer Relevanz, da indirekt Inspektionsziele darin formuliert sind.

Auch die Art der Wissensmanagement Systeme ist für deren Inspektion von Wichtigkeit. Deswegen werden im nächsten Abschnitt die verschiedenen Typen von Wissens-Management Systemen behandelt.

3.9 Bausteine des Wissensmanagements

In Anlehnung an Probst [Pro97], werden hier die Bausteine des Wissensmanagements vorgestellt. Probst hat dieses Konzept aus intensiver Zusammenarbeit mit verschiedenen Unternehmen entwickelt. Er hat versucht die Theorie und die Praxis so nah wie möglich zusammenzubringen.

Die Wissensbausteine von Probst stellen Kernprozesse dar, die alle miteinander verbunden sind.

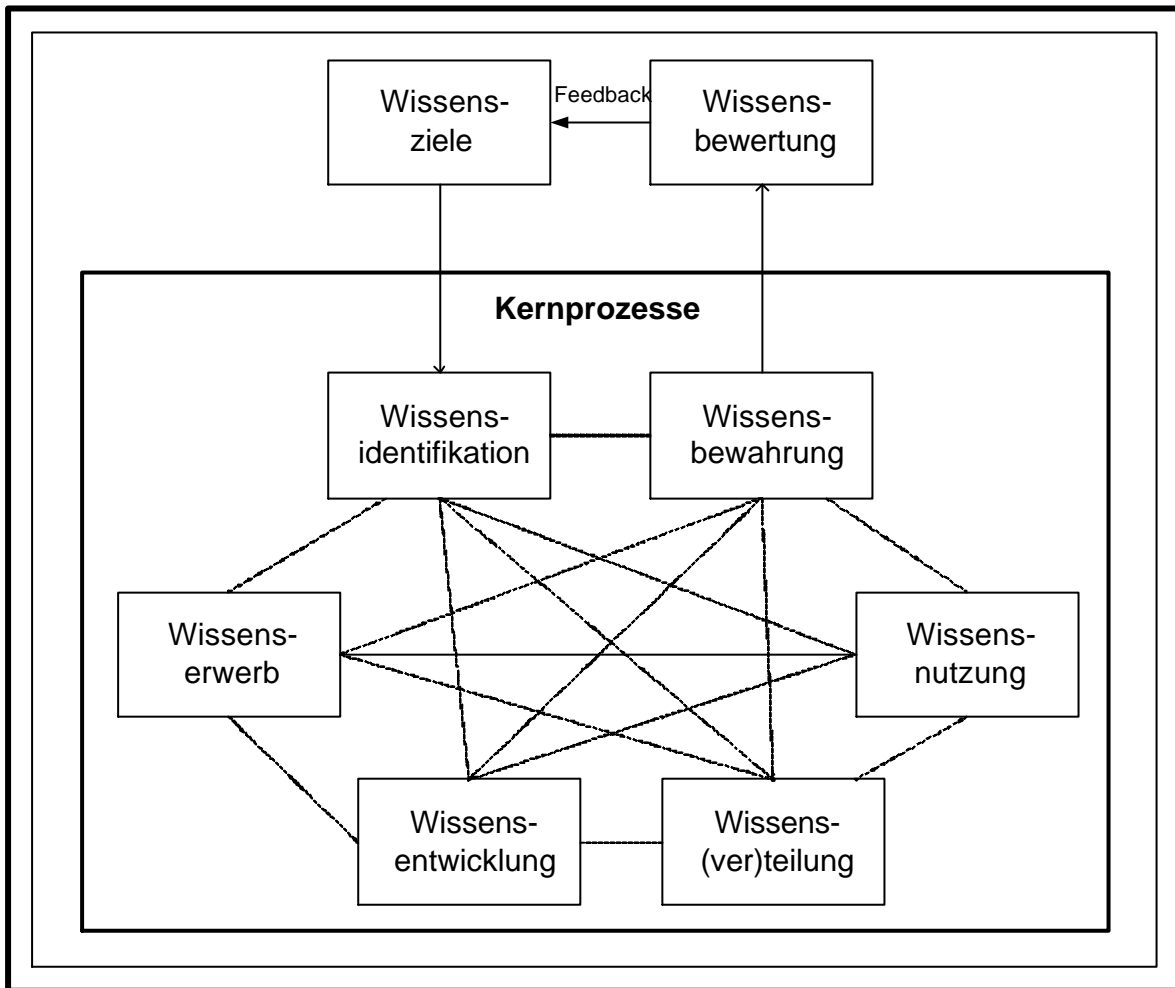


Abbildung 19 Bausteine des Wissensmanagements

3.9.1 Wissensidentifikation:

Die Wissensidentifikation beschäftigt sich mit der Frage, wie intern und extern Transparenz über vorhandenes Wissen geschaffen werden kann. Der Bestand an internen und externen Daten, Informationen und Fähigkeiten muss für alle sichtbar und verfügbar sein, so dass es die Mitarbeiter bei allen Suchaktivitäten unterstützen kann.

3.9.2 Wissenserwerb:

Ein Unternehmen kann nicht alles alleine entwickeln, was er an Fähigkeiten und Kompetenzen braucht, um konkurrenzfähig zu bleiben. Deswegen kaufen sich viele Unternehmen Experten ein, oder externes Know-how von anderen Firmen. Dieser Baustein des Wissensmanagements muss daher systematisch miteinbezogen werden bei dem Prozess der Umsetzung des Wissensmanagements.

3.9.3 Wissensentwicklung:

Dieser Baustein steht in einer komplementären Beziehung zum Wissenserwerb. Dabei geht es darum, neue Fähigkeiten, neue Produkte, bessere Ideen und leistungsfähigere Prozesse zu produzieren. Es handelt sich bei der Wissensentwicklung um alle Managementaktivitäten, die dazu beitragen, dass eine Organisation sich mit der Produktion neuer internen und externen Fähigkeiten. Es werden nicht nur die traditionellen Aspekte der Wissensentwicklung wie in der Forschung, Entwicklung oder Marktforschung, vielmehr zielt es bei der Wissensentwicklung um Mitwirkung aller Bereiche einer Organisation, nämlich auch die Bereiche die nur für die Leistungserstellung relevant waren.

3.9.4 Wissens(ver)teilung:

Bei diesem Prozess, geht es darum, isoliert vorhandene Informationen oder Erfahrungen für das gesamte Unternehmen verwendbar zu machen. Diese Verteilung von Erfahrungen ist für ein Unternehmen unabdingbar. Allerdings sollte man dem nicht die überhand geben, und unnötig Zeit, Geld und Effizienz verlieren, nur der Verteilung zugunsten. Denn wichtig dabei sind folgende Fragen:

- Wer sollte was wissen?
- In welchem Umfang soll dieses Wissen für die betroffene Person verfügbar sein?
- Wie kann der Prozess der Wissensverteilung erleichtert werden?

Diese Fragen können sinnvoll nach dem ökonomischen Prinzip beantwortet werden. Denn, es muss nicht alles von jedem gewusst werden. Daher beinhaltet dieser Prozess eine sinnvolle Beschreibung und Steuerung des Wissensverteilungsumfanges. Es wird dabei der Übergang von Wissensbeständen von der individuellen Ebene auf die Gruppen- und Unternehmensebene analysiert.

3.9.5 Wissensnutzung:

Ziel und Zweck von Wissensmanagement ist der produktive Einsatz des eigenen Wissens einer Organisation zum Nutzen des Unternehmens. Dieser Prozess wird mit Wissensnutzung beschrieben. Es genügt leider noch nicht für die Wissensnutzung, dass im Unternehmen die Prozesse der Wissensidentifikation und Verteilung zentraler Wissensbestände erfolgreich ablaufen. Dies garantiert leider nicht, dass die Wissensbestände alltäglich genutzt werden. Denn es wird eher auf bewährte Routinen im Arbeitsalltag zugegriffen, als auf neue Ideen auszuweichen, deren Lernaufwand zum einen groß ist, und zum anderen deren Umsetzungserfolg nicht mal sicher ist. Es muss daher eine intensive Überzeugungsarbeit beim Nutzer geleistet werden, damit es überhaupt zu einer Umsetzung innovativer Ideen kommt. Hilfreich bei der Wissensnutzung können zusätzlich zum Abschaffen von Wissensbarrieren auch Anreize für eine aktive Wissensnutzung sein, wie beispielsweise eine Erhöhung der Kompetenzen und des Handlungsspielraumes oder gar eine Vergeltung nach dem Wissensumschlag.

3.9.6 Wissensbewahrung:

Hier ist die zentrale Frage, wie kann ich mein erworbenes Wissen vor Verlusten schützen? Denn einmal erworbene Fähigkeiten stehen nicht automatisch für die Zukunft zur Verfügung. Es müssen Managementstrategien eingesetzt werden für die Erhaltung von Erfahrungen, Informationen und Dokumenten. Hierbei muss auf die effiziente Nutzung von den unterschiedlichsten Speichermedien eines Unternehmens für Wissen. Dabei muss die Frage analysiert werden: welches Wissen soll eigentlich bewahrt werden? Denn unnötiges Wissen stellt nur ein Hindernis dar, beim Suchen nach

brauchbarem Wissen. Auch auf die aktuelle Gültigkeit des Wissens muss geachtet werden, denn veraltetes Wissen kann verheerende Konsequenzen haben bei einer logischen Entscheidung. In diesem Zusammenhang spricht Probst [Pro98] von der Wichtigkeit vom Verlernen des Wissens. Daher muss Wissen nicht nur bewahrt werden, sondern vorher schon nach präzisen Kriterien ausgewählt, angemessen gespeichert und regelmäßig aktualisiert werden.

3.10 Typen von Wissensmanagement Systeme

Die Wissensmanagement Systeme unterscheiden sich in der Art und Weise wie sie aufgebaut werden. Allerdings sind für alle Wissensmanagement Systeme Ontologien sehr nützlich. Ontologien sind definierte Begriffsstrukturen. Sie werden unterteilt je nach Anwendungsbereich oder Unternehmen. Beispielsweise werden sie nach Aktivitäten, Zustand, Organisation, Zeit, Ressource, Produkt oder Service gegliedert. Aus der Recherche ergeben sich folgende Kategorien:

- Die alle erste Kategorie sind die Nicht-Computerbasierten Wissensmanagement Systeme.
- Die Dokumentenbasierten Wissensmanagement Systeme beinhalten alle Formen von Content Management Systeme. Diese sind:
 - Projektdokumente (Spezifikationen, Entwurfsdokumente, Testdokumente, Vertragsunterlagen)
 - Standard-Referenz-Werke in einem Bereich
 - Visuelle Dokumente (Fotos, eingescannte Pläne, Ikonische Dokumente)
 - Technische oder wissenschaftliche Berichte
 - Bücher, Dissertationen, Normen, Archivdokumente, Führer, Dossiers usw.

- Digitale Bibliotheken
- Online-Dokumentationen, Benutzerhandbücher, Referenz-Handbücher usw.
- Dann gibt es die Wissensbasierte Systeme, die auf die Formalisierung von Wissen aufbauen.
- Eine weitere Kategorie sind die Fallbasierten Wissensmanagement Systeme. Sie bauen auf formalisierte Fallformate und erlauben eine kontinuierliche Evolution. Dabei sind Fälle Problem/Lösungspaare von vergangenen Erfahrungen. Diese werden gespeichert und dem Benutzer in der Fallbasis zur Verfügung gestellt.
- Die verteilten Wissensmanagement Systeme kombinieren verschiedene heterogene Quellen, wie Agenten oder Web-Suchmaschinen.
- Man kann natürlich ein Wissensmanagement System auch kombiniert aufbauen, in dem man zwei oder mehr Strategien zusammenmischt.

Diese Vielfalt der Wissensdokumente warnt uns im Voraus, dass wir auf eine individuelle Behandlung bei der Konzeption und Entwicklung von Inspektionsmethoden der verschiedensten Dokumente achten sollen.

3.11 Zusammenfassung und Fazit

Dieses Kapitel handelte von Wissen und Wissensmanagement. Zuerst wurde „Wissen“ auf vielen Ebenen definiert. Danach wurde der Anreicherungsprozess vorgestellt, der den Übergang aus Daten zu Information zu Wissen beschreibt. Angeleitet von der epistemologischen Definition, die das Wissen in Metawissen, latentes, implizites und explizites Wissen aufteilt, habe ich es für wichtig gehalten, auch das HANSE-Modell von Snowden vorzustellen. Dieses besagt, dass Wissen nicht immer in reiner expliziter oder impliziter Form vorzufinden ist.

Wie wird aber Wissen akquiriert? Dieser Frage wurde auch nachgegangen, in dem die Wissensspirale von Nonaka vorgestellt wurde. Danach wurden die Wissensträger beschrieben, bevor die Wissensdarstellungsformen vorgestellt wurden.

Anschließend wurde das Wissensmanagement Konzept vorgestellt. Dabei habe ich mich auf die Bausteine des Wissensmanagement von Probst eingeschränkt. Zum Schluss habe ich noch einige der verschiedenen Typen von Management Systeme vorgestellt.

Dieses Kapitel bildet eine weiter wichtige Säule für diese Diplomarbeit. Daraus haben wir die Probleme bei der Darstellung von Wissen kennengelernt, daraus können wir die Forderungen an Wissensdokumenten ableiten, die die Qualitätsmerkmale darstellen sollen, wie Klarheit oder Konsistenz.

Auch aus den verschiedenen Wissensformen, wurde das Fazit abgeleitet, dass für eine Wissensinspektion nur explizites Wissen von Bedeutung sein kann.

Die Wissensinspektion lässt sich in das Konzept des Wissensmanagements von Probst in folgende Bausteine einbetten:

- Bei der Wissensbewertung, um Inspektionsbasierte, qualitativ hochwertige Aussagen über das Wissen machen zu können.
- Bei der Wissensidentifikation, um Aussagen darüber zu machen, wie transparent das interne oder extern erworbenes Wissen ist.
- Beim Wissenserwerb, um das extern erworbene Wissen zu inspizieren.
- Und bei der Wissensentwicklung, um eine ständige Kontrolle über die neuproduzierten Fähigkeiten, Produkte und leistungsfähigere Prozesse zu gewährleisten.

Nach diesen Erkenntnissen können wir getrost die Konzeption und Entwicklung von Wissensinspektionen angehen. Dies ist Gegenstand vom nächsten Kapitel. Darüber hinaus beinhaltet das nächste Kapitel

die Durchführung des Inspektionsprozesses und seine ausführliche Evaluierung.

4 Konzeption und Entwicklung der Inspektionsmethoden

Dieses Kapitel beschäftigt sich mit der Konzeption und Entwicklung von Lesetechniken zur Inspektion von Wissensdokumenten und Erfahrungen. Die im Kapitel zwei eingeführten Inspektionsmethoden im Software Engineering werden auf ihre Tauglichkeit für die Inspektion von Wissen und Erfahrungen hin untersucht. Vorerst werde ich aber die Frage nach den Inspektionszielen für Wissen anschnitten. Diese Frage zieht mit sich auch die Überlegung und Definition der relevanten Merkmale eines Wissensdokumentes. Danach wird das durchgeführte Experiment vorgestellt. Anschließend werden die Ergebnisse der Evaluation angegeben.

4.1 Die Konzeption der Inspektionsmethoden für Wissen und Erfahrungen

Wissensdokumente und Erfahrungsberichte werden von Menschen entwickelt und geschrieben. Da der Mensch fehlbar ist, werden diese Dokumente mit Sicherheit auch fehlerbehaftet sein. Sie weisen Eigenarten auf, wie Ungenauigkeit, keine Struktur oder Kontextabhängigkeit. Um diese Problem und Fehler zu beheben und somit die Qualität eines Wissensdokumentes zu steigern, sollen Inspektionsmethoden speziell für Wissensdokumente entwickelt werden. Diese Methoden sollen die Merkmale und Schwierigkeiten eines Wissensdokumentes berücksichtigen.

Die Elemente, die inspiziert werden können, müssen in greifbarer Form sein. Dies ist der Fall nur beim Expliziten Wissen, wie wir im letzten Kapitel gesehen haben. Dazu werden hier folgende Fragen gestellt:

- Was sind die typischen Probleme eines Wissensdokumentes?
- Was für Qualitätsmerkmale sollen von einem Wissensdokument gefordert werden?
- Was sind die Ziele einer Wissensinspektion?

- Was sind die Vorteile einer Inspektion?

Die Antworten zu diesen Fragen, die im Laufe der Konzeptionsphase dieser Diplomarbeit entstanden sind, werden in den folgenden Abschnitten bereitgestellt.

4.1.1 Probleme eines Wissensdokumentes

Die Wissens Elemente haben Problembereiche und weisen daher Schwierigkeiten auf, die für die Entwicklung von Inspektionsmethoden berücksichtigt werden müssen. Nämlich, dass die Wissensdokumente:

- Meistens unstrukturiert sind: Erfahrungsberichte, Wissensdokumente weisen meistens keine bzw. wenig Struktur auf. Sie werden von den Wissensträger Freihand dokumentiert, und somit werden sie individuell gestaltet.
- Schwer zu verstehen sind: da die Wissensdokumente von Fachleuten geschrieben werden, sind sie für Dritte nur schwer verständlich. Denn diese Experten schreiben nicht alles im Detail, und können daher für Dritte und sogar für Leute relevante Informationen weglassen.
- Kontextabhängig sind: Wissensdokumente entstehen in einem gewissen Kontext und sind meistens nur schwer davon entkoppelbar.

Diese Probleme stellen Unterschiede zu Softwaredokumenten wie Anforderungen, Entwürfen und Code. Diese Art von Dokumenten unterliegt meistens einer klaren Struktur. So werden Anforderungen von allgemeinen Kundenbeschreibung in Kategorien umgeschrieben, dabei werden sie in funktionale und nicht funktionale Anforderungen klassifiziert.

4.1.2 Qualitätsmerkmale eines Wissensdokumentes

Nachdem wir nun die Probleme der Wissens Elemente kennen gelernt haben, widmen wir uns nun ihre Qualitätsmerkmale. Die Idee dabei

ist, dass diese Qualitätsmerkmale Anforderungen stellen, die überprüft werden können. Dabei wird erwünscht, die Probleme eines Wissensdokumentes klein zu halten bzw. zu beheben.

Inspektionen von Wissensdokumenten und Erfahrungen haben zum Ziel, die Qualität solcher Dokumente zu steigern. Dies geschieht in dem die Dokumente auf ihre Qualitätsmerkmale hin überprüft werden, am Besten soll das geschehen, sobald ein Dokument fertig erstellt ist. Nun stellt sich die Frage, was für Qualitätsmerkmale ein Wissensdokument aufweisen soll?

Um die erwünschten Qualitätsmerkmale eines Wissensdokumentes zu entwickeln, wurde hier auf die Wissensmodellierungsformen Bezug genommen. Die Modellierung von Wissen geschieht, wie wir in Kapitel 3 gesehen haben, meistens ungenau, unsicher, unvollständig, widersprüchlich oder prototypisch. Um diesen Problemen entgegenzuwirken wurden folgende Forderungen an ein Wissensdokument gestellt, die dann seine Qualitätsmerkmale darstellen sollen:

- Klarheit:
damit ist gemeint, dass die Angaben, die in einem Wissensdokument gemacht sind, klar und eindeutig sein sollen. Dadurch wird die Verständlichkeit des Dokumentes erheblich erleichtert. Was der Ungenauigkeit entgegenwirken soll.
- Vollständigkeit: Ein Wissensdokument soll alle nötigen Elemente enthalten, die für die Anwendung oder Anpassung der Lösung notwendig sind. So wird überprüft, ob und in wie weit ein Wissensdokument unvollständig modelliert wurde.
- Konsistenz:
Alle im Wissensdokument gemachten Angaben sollen sich nicht widersprechen und nicht mehrdeutig sein. Dieses Merkmal soll dann die Inspektion auf Widersprüchlichkeit ermöglichen.

- **Korrektheit:**
Hierbei geht es darum, dass die Angaben, das Problem, die beschriebene Lösung und der gesamte Kontext korrekt beschrieben sein sollen. Damit wird erwünscht, dass das Dokument nach einer Inspektion keine unsicheren oder gar falschen Informationen beinhaltet.
- **Funktionalität:**
Bei diesem Merkmal geht es darum, ob die beschriebenen Angaben notwendig und ausreichend sind, um die Ziele des Wissensdokumentes zu erreichen.
- **Testbarkeit:**
Dabei soll überprüft werden, ob die enthaltenen Informationen getestet, analysiert oder vorgeführt werden können, sodass sie gegen die Anforderungen verifiziert werden können.

4.1.3 Vorteile durch Wissensmanagement

Die Verwendung von Wissen, besonders von qualitativ überprüfem Wissen, führt unmittelbar zu spürbaren Vorteilen. Dabei wird nämlich erwünscht, dass:

- der Benutzer dieser Wissens Elemente bestens bei seinen täglichen Aufgaben unterstützt wird,
- falls ähnliche Aufgaben oder Projekte wieder auftreten diese schneller bearbeitet werden können ,
- Die Fähigkeiten des Benutzers selbst verbessert werden, in dem er von diesen Wissens Elementen lernt, und von der Art und Weise wie ein vergangenes Problem gelöst wurde, besonders bei Problem/Lösungsfällen,
- Die Kosten erheblich sinken, da wiederauftretende Probleme und Erfahrungen schneller bearbeitet werden.

4.2 Wissensinspektionen

4.2.1 Ziele einer Wissensinspektion

Genau wie bei der Softwareinspektion, wollen wir Wissensinspektionen aus gewissen Zielen heraus durchführen.

Die Ziele einer Wissensinspektion sind:

- Fehler in Wissensdokumenten entdecken, identifizieren und beschreiben zu können.
- Die Wiederverwendbarkeit der Lösungen/Erfahrungen eines Wissensdokumentes zu überprüfen.
- Verbesserung der Qualität der Wissens Elemente.
- Sichtbare Fehlerreduktion zu erzielen.
- Daten über den Wissensmanagementprozess zu sammeln, beispielsweise Daten über Wiederverwendbarkeit und Aufwand.
- Eine Wettbewerbsfähigkeit oder gar einen Vorsprung durch qualitativ hochwertige Wissensdokumente auf den Markt zu erreichen.

4.2.2 Vorteile einer Wissensinspektion

Wissensinspektionen führen zu qualitativen Vorteilen. Diese können sein:

- Die frühen Entdeckung von Fehlern,
- Bessere Struktur von Wissensdokumenten,
- Verbesserung der Wissensdokumente,
- Verbesserung der Lösungsfälle und Erfahrungen,
- Erhöhter und effizienter Lerneffekt,
- Erhöhte Zufriedenheit des Benutzers.

Nachdem nun die Ziele, die Problembereiche und die Vorteile einer Wissensinspektion stehen, lauten die nächsten zwei Schritte wie folgt:

- Auswahl der Inspektionsmethoden: aus den wohldefinierten Inspektionsmethoden im Software Engineering werden drei Methoden ausgewählt, die den Anschein haben, sich zum Inspizieren von Wissens-elementen gut zu eignen.
- Verfeinerung und Anpassung der ausgewählten Prüfverfahren: die drei ausgewählten Methoden sollen verfeinert werden und an die Bedürfnisse der Wissens-elemente angepasst werden. Dabei sollen die oben erwähnten Merkmale, Eigenschaften und Probleme von Wissens-elementen berücksichtigt werden.

4.3 Auswahl und Entwicklung der Inspektionsmethoden

Um geeignete Inspektionsmethoden für die Wissensinspektion auszuwählen, wurden alle im Kapitel zwei vorgestellten Techniken untersucht und auf ihre Tauglichkeit für die Bedürfnisse von Wissensdokumenten hin überprüft.

Zuerst sei hier festgehalten, dass der Entwicklung der drei Methoden die Idee zu Grunde lag, dass wir die gemeinsamen Qualitätsmerkmale eines Wissensdokumentes inspizieren wollen. Hier sollen keine spezifischen Dokumente betrachtet werden, sondern eher allgemeingültige Methoden konzipiert und angeboten werden, die später auf die meisten Arten von Wissensdokumenten anwendbar sind.

4.3.1 Probleme bei Wissensinspektionen

Bei der Betrachtung der Softwareinspektionsmethoden, wurde aufgefallen, dass sich einige Methoden von vorneherein nicht eignen, da sie für spezielle Dokumente entwickelt worden sind wie z.B. das Lesen durch Schrittweise Abstraktion, wie wir gleich begründen werden. Da aber unsere Wissensdokumente vielfältig sind, muss

Ausschau gehalten werden nach Methoden, die sich für die meisten Dokumente eignen.

Die Methoden, die bei genauer Betrachtung gleich durchfielen, da sie Einschränkungen auf für spezielle Dokumenttypen aufweisen, sind:

1. Kontrollflussorientiertes Lesen: Diese Lesetechnik ist wie in Kapitel 2 beschrieben speziell für Kontrollfüße konzipiert worden. Da wir aber hier allgemeine Wissensdokumente inspizieren wollen, kommt diese Technik nicht in die Auswahl. Lesen durch schrittweise Abstraktion: auch hier handelt es sich um eine Technik speziell für Code-Dokumente. Aus den selben Gründen wie oben kommt auch diese Lesetechnik nicht in Frage.
2. Aktive Design Review: Diese Inspektionsmethode ist auch nur für Design-Dokumente geeignet. Daher fällt sie auch bei der Auswahl durch.
3. Function Point basierte Lesetechniken sind für Anforderungsdokumente entwickelt worden. Diese Einschränkung macht sie dann auch nicht zum Kandidaten für diese Arbeit.

Dagegen lassen sich folgende Techniken mit einigen Anpassungen gut für Wissensinspektion einsetzen:

1. Walkthroughs: wenn der Autor dabei ist und den Inspektoren durch das Dokument führt.
2. Reviews: wenn dabei wirklich Wissensprodukte entwickelt werden, die nach verschiedenen Phasen immer wieder überprüft werden müssen
3. Audits: wenn die Spezifikationsdokumente, zu denen Abweichungen des Wissensdokumentes festgestellt werden sollen, vorhanden sind.

4.3.2 Anforderungen an eine Wissens-Inspektionsmethode

Nach den im obigen Abschnitt gewonnenen Erkenntnissen, standen die folgenden drei folgenden Lesetechniken zur Auswahl. Da das Ziel eine bewährte Methodik wie die Perspektiven basierte Lesetechnik in einem Experiment auch gegenüber anderen Methoden und der Standardtechnik „Ad-hoc“ zu vergleichen, wurden folgende drei Methoden in die Auswahl genommen:

4. Ad-hoc Inspektion: dieses Standardvorgehen ist einfach und intuitiv und kann ohne Anpassungen auch für Wissensdokumente genommen werden.
5. die Checklisten basierte Lesetechnik: dabei sollen die Checklisten, also die Fragen unter Berücksichtigung der Probleme und erwünschten Merkmalen eines Wissensdokumentes entwickelt werden.
6. Die Perspektiven basierte Lesetechnik: bei dieser Methode müssen nicht nur die in den Szenarien enthaltene Checklisten angepasst und verfeinert, vielmehr müssen die Perspektiven und die darin enthaltenen Aktivitäten entwickelt werden.

4.3.3 Entwicklung der Methoden

Nach der Auswahl der drei Methoden geht es hier darum, diese für die Wissensdokumente maßzuschneidern.

4.3.3.1 Ad-hoc

Die Ad-hoc Inspektion ist ein einfaches und intuitives Vorgehen, bei dem die Inspektoren im Team das Wissensdokument überprüfen sollen. Hier ist kein Raum geboten, Einfluss auf die Arbeit der Inspektoren zu nehmen. Man kann lediglich, ausgehend von den erwünschten Qualitätsmerkmalen eines Wissensdokumentes Beispielfragen ableiten, die den Inspektoren als Hinweis dienen. Dabei wird den Inspektoren klar gemacht, worauf sie beim Inspizieren des Dokumentes achten sollen. Der Experiment-Abschnitt gibt dazu einige Beispiele.

4.3.3.2 Checklisten basierte Lesetechnik

Das Ziel dabei ist, Checkliste zu entwickeln, die den speziellen Bedürfnissen der Wissensdokumente genügen. Hierzu werden Fragen an die Merkmale und Probleme eines Wissensdokumentes entwickelt.

Bei der Konzeption von Checklisten, müssen die zu inspizierenden Dokumente untersucht werden. Dabei würde man den Typ des Dokumentes feststellen. Danach müssen die Fragen entwickelt werden. Diese werden von erfahrenen Experten unter Berücksichtigung des Dokumenttyps gestellt. Dabei entstehen die Fragen meisten aus vergangenen Erfahrungen mit Inspektionen von ähnlichen Dokumenten oder Erfahrungen aus ähnlichen Projekten.

Bei unseren Wissens-elementen müssen die Fragen auf unterschiedliche Kategorien gerichtet werden, so dass mehr Struktur, Verständnis und Klarheit in das Wissensdokument reingebracht werden. Beim Betrachten von Wissensdokumenten aus einer Quelle, fällt auf, dass alle Wissensdokumente zwei Gemeinsamkeiten aufweisen bezüglich der Struktur die sie darstellen, und den Inhalt, den sie weitergeben. Daher wurde die Idee verfolgt, die Fragen nach Syntax (Struktur) und Semantik (Semantik) zu differenzieren. Da ein Wissensdokument wie bereits erwähnt meistens nicht klar genug ist, nicht strukturiert und zu allgemein ist, sollte diese Unterteilung den Studenten eine willkommene Hilfe sein. Wie die Checkliste dann schließlich aussahen, finden Sie im nächsten Abschnitt über CBR.

4.3.3.3 Perspektiven basierte Lesetechnik

Bei dieser Inspektionsmethode werden Rollen und Szenarien für die Wissensinspektion abgeleitet. Hierzu werden die Dokumente in ihren Gemeinsamkeiten betrachtet. Gemeinsame Rollen sind auf jeden Fall folgende:

- Autor eines Wissensdokumentes, dabei soll hauptsächlich überprüft werden, ob das Wissensdokument eine klare Struktur (Syntax) besitzt.

- Leser eines Wissensdokumentes, der zur Hauptaufgabe hat, das Wissensdokument auf einen klaren, verständlichen und widerspruchsfreien Inhalt untersucht.
- Benutzer eines Dokumentes. Hier besteht der Unterschied zum Leser darin, dass der Leser eines Dokumentes dessen Inhalt nicht anwenden muss. Daher gehört noch zu den Aufgaben eines Benutzers, das Dokument auf seine Anwendbarkeit zu überprüfen.
- Tester eines Dokumentes, der das Dokument z.B. in seiner Richtigkeit in einem realen Kontext testet.

Die Szenarien beinhalten in Anlehnung an die PBR-Lesetechnik im Software Engineering auch hier Anweisungen und Aktivitäten, die die Inspektoren in ihren jeweiligen Rollen befolgen sollen. Es wurde darauf Acht gegeben, dass die Rollen erklärt sind und die Aktivitäten, die die Inspektoren tätigen sollen, verständlich sind. Auch hier wurde versucht, diese Szenarien so allgemein wie möglich zu halten, so dass sie für verschiedene Wissensdokumente ohne große Anpassung genommen werden können. Die Rollen und Szenarien finden Sie in der praktischen Erklärung im nächsten Abschnitt.

Bei der **Entwicklung von den Szenarien** wurde als erste eine Stakeholderanalyse betrieben, wie bei der PBR-Methode im Software Engineering[Lai00]. Dabei wurde das Wissensdokument auf seine Interessensträger hin analysiert. Aufgabe war die Perspektiven zu ermitteln, die für das Wissensdokument von Bedeutung sind. Dabei sind u.a. die oben beschriebenen Rollen entstanden.

Die **Entwicklung der Fragen** zu den Checklisten geschieht so, in dem systematisch die Aufgaben der Rollen genau untersucht werden. Dabei entstehen Schwerpunkte, die auf jeden Fall aus Sicht der jeweiligen Rolle. Bei der Entwicklung der Fragen müssen die Qualitätsmerkmale und die Problembereiche des Dokumentes miteinbezogen werden, die vorher herausgefunden werden müssen. Bei diesem Experiment dienten die Qualitätsmerkmale, die Problembereiche und die aus Kapitel 3 gewonnenen Erkenntnissen zum Wissen und Wissensfragen als Grundlage zur Entwicklung der Checklisten.

Die speziellen Fragen einer Checkliste führen natürlich dazu, dass die Gutachter in die jeweilige Perspektive versetzt werden, so dass die persönliche Erfahrung des einzelnen Gutachters, mit dem selben Projekt oder mit anderen Projekten kaum noch eine Rolle spielt.

Dadurch, dass die Entwicklung der Szenarien und Checklisten für die PBR viel Zeit in Anspruch genommen hat, im Vergleich zur Entwicklung von Checklisten bei der CBR, wird erhofft, dass gerade für diese Inspektion viel weniger Zeitaufwand von unseren Inspektoren in Anspruch genommen wird.

4.4 Der Aufbau des Experimentes

Das Experiment wurde in freundlicher Zusammenarbeit mit den Hauptstudiumsstudenten der Vorlesung "Erfahrungsmanagement – Entwicklung und Einführung" von Dr. Althoff durchgeführt. Dazu standen uns ca. zwei Stunden zur Verfügung.

Das Experiment umfasste drei große Teile:

- Ein Einführungs- und Motivationsteil, das den Studenten die ganzen Vorzüge einer Inspektion klar machen sollte und die Ziele, die wir erreichen möchten bei einer Wissensinspektion in Anlehnung an die Software Inspektionsverfahren.
- Ein zweites Teil, in dem wir auf die drei ausgewählten Inspektionsmethoden ausführlich eingegangen sind und wo die Studenten die Gelegenheit hatten, Verständnisfragen zu jeder Methode zu stellen.
- Im dritten Teil erhielten die Studenten dann die Wissensdokumente und führten die Inspektionsmethoden durch, eine nach der anderen.

Das Experiment sollte zwei Stunden dauern. In der ersten Stunde des Experimentes wurden die theoretischen Grundlagen der Inspektionsverfahren behandelt während die zweite Stunde zur praktischen Ausführung der drei verschiedenen Inspektionsverfahren den Studenten zur Verfügung stand.

4.4.1 Einführung

Als Einstiegsmotivation dienen die Ziele einer Inspektion und den Vergleich von Wissensinspektionen mit den Case based Reasoning(CBR) Fälle. Da die Studenten sich während der Vorlesung "Erfahrungsmanagement – Entwicklung und Einführung" sich u.a. auch mit CBR befasst haben, ist es sinnvoll diesen Vergleich einzubringen. Die CBR-Fälle, die ja die Problem/Lösungspaare darstellen, sollen dazu dienen, beim Auftreten von ähnlichen Problemen, eine Lösungsmöglichkeit bereit zu stellen. Das Problem und somit die Lösung können und müssen nicht immer hundertprozentig übereinstimmen. Aber jede aus einem vorigen ähnlichen Fall gewonnene Erfahrung beschleunigt den Lösungsprozess. Dieser Grundidee trifft genau unsere Vorstellungen von der Wissensinspektionen. Beispiele von Wissensdokumente sind vielfältig und können Erfahrungen sein wie z.B. Projekterfahrungen, oder treten in Form von Problemfällen oder Situationsbeschreibung.

Die Probleme, die solche Wissensdokumente beinhalten werden den Studenten auch erklärt, nämlich dass die Wissensdokumente meistens nicht leicht verständlich sind, kontextabhängig und nicht strukturiert sind.

Nicht desto Trotz müssen die Studenten die wesentlichen Ziele, die wir gemeinsam erreichen wollen bei einer Wissensinspektion sich stets vor Augen halten. Diese sind eben die Verbesserung der Qualität solcher Wissensdokumente sowohl im Verständnis und in der Struktur, und die qualitativ bessere Unterstützung des Benutzers bei den immer wiederkehrenden Aufgaben. Dadurch entwickelt sich das „Wissen“ der einzelnen Mitarbeiter auch weiter. Hier sei auf Kapitel drei hingewiesen, das sich ausführlicher mit diesen Themen befasst hat.

Wir haben auch Wert darauf gelegt, den Studenten die Vorzüge der Inspektionsverfahren in der Software vorzustellen, in Form von klaren Aussagen, die auf die Ergebnisse der Anwendbarkeit von Inspektionsverfahren in den letzten Jahren im Bereich des Software Engineerings gemacht wurden. Diese sind u.a. :

- Verbesserung der Qualität der Software Produkte

- Sichtbare Fehlerreduktion
- Kosteneinsparung

Darüber hinaus wurde auf das Inspektionsverfahren allgemein eingegangen. Hier wurde das Vier-Phasen Modell der Software Inspektion vorgestellt, das aus einer Planungsphase, einer Fehlererkennung-, Kollektions- und Korrekturphase besteht, wie unteres Bild zeigt:

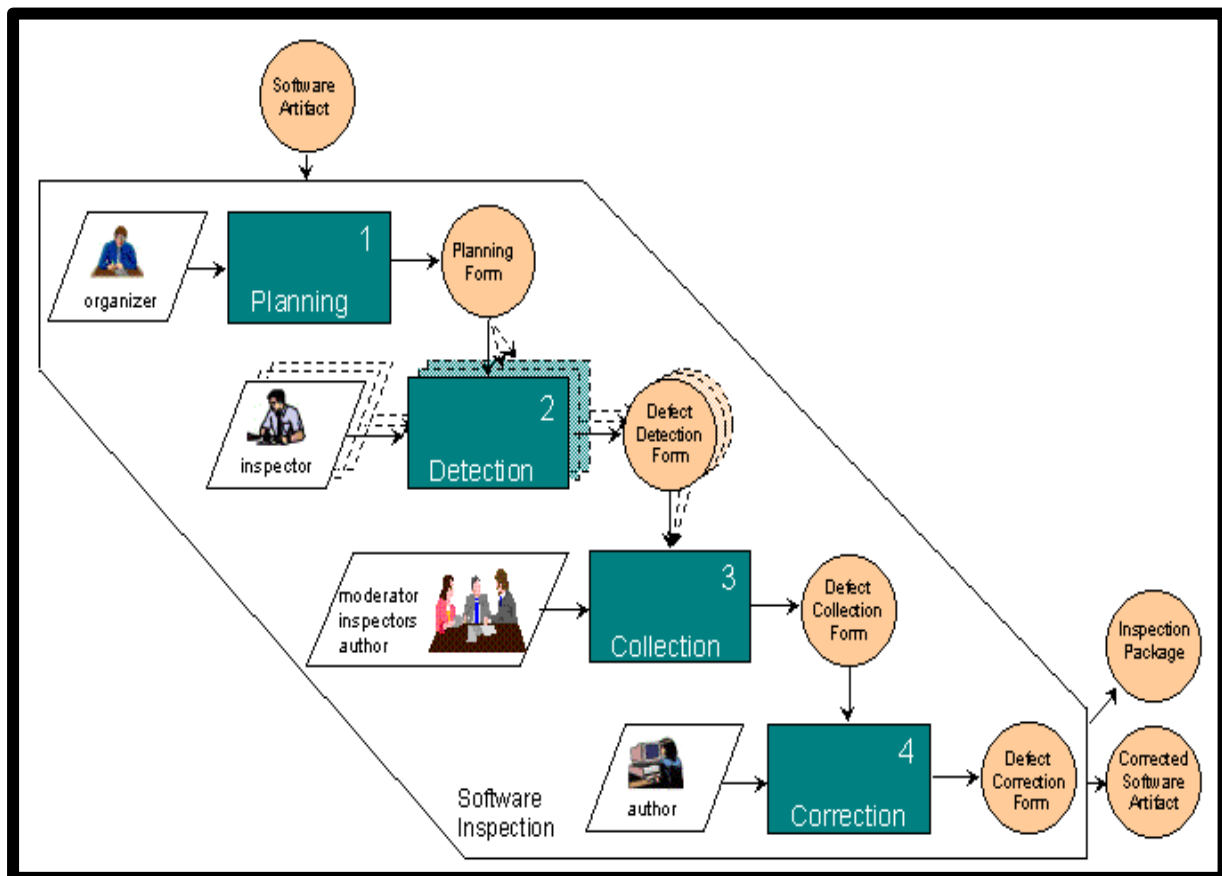


Abbildung 20 die vier Phasen einer Inspektion

Hier sei betont, dass die Detection Phase, also die Fehlerfindungsphase ausführlicher behandelt worden ist, damit die Studenten eine bessere Vorstellung davon kriegen, wie sie überhaupt Fehler finden können, und vor allem, dass sie darauf hingewiesen werden, dass ihre Rolle als Inspektor nur die Aufgabe vorsieht, den Fehler zu finden und nicht zu korrigieren. Denn das Korrigieren eines

Defektes liegt je nach Methode in der Obhut des Autors. Für die ausführlicheren Erläuterungen sei hier auf Kapitel 2 hingewiesen.

Außerdem wurden die sechs Defektklassen eines Softwarefehlers vorgestellt. Zu jeder Klasse wurde noch ein Beispiel wie der einer Klasse zugehörigen Fehler aussehen kann und eine mögliche Auswirkung dieses Fehlers, wie die Tabelle 1 anschaulich macht.

Omission (O)	Necessary information about the system is missing, e.g., the interface to an external entity is not specified.
Ambiguous Information (AI)	Information within the documentation is ambiguous /unclear, i.e., various interpretations may be derived from the requirements by the designer.
Inconsistency (IC)	Information within one part of the documentation is inconsistent with other information at another part.
Incorrect fact (IF)	Some statement contained in one part of the documentation asserts a fact that contradicts common sense or domain knowledge.
Extraneous (E)	Information is provided that is not needed or used.
Miscellaneous (M)	Other defects.

Tabelle 1

Fehlerklassen

Nachdem die Studenten die verschiedenen Defektklassen kennen gelernt haben, wurden sie mit den Aufgaben eines Inspektors vertraut gemacht, da sie schließlich in Kürze darauf selbst zu Inspektoren werden und wissen müssen, wie sie vorzugehen haben. Hierzu wurden sie nicht mit unnötigen Informationen zugeschüttet, sondern sie erhielten einfache Anweisungen, die den Kern der Inspektion widerspiegeln. In diesem Rahmen wurden die Studenten aufgefordert:

1. Die angegebenen Anweisung der jeweiligen Inspektionsmethode zu befolgen. Diese sind bei jeder Methode angegeben und reichen von einfachen Schritten bei

der Ad-hoc Methode bis hin zu einem richtigen Szenario, in das sich der einzelne Inspektor reinversetzen soll und das einer bestimmten Inspektionsrolle entspricht.

2. Die gefundenen Fehler in dem Wissensdokument zu markieren und dann in das dafür vorgesehene Defektformular einzutragen, mit einer Defektnummer, die angefangen bei 1 jedem Fehler zugewiesen werden muss, der Platzierung des Fehlers mit der Zeilennummer und Seitennummer, mit dem Fehlertyp entsprechend der Defektklassen Tabelle und einer Beschreibung des Fehlers und des damit verbundenen Problems.
3. jedem Fehler eine Defektklasse zuzuweisen, die Platzierung des Fehlers mit der Zeilennummer und Seitennummer anzugeben, mit dem Fehlertyp entsprechend der Defektklassentabelle und einer Beschreibung des Fehlers und des damit verbundenen Problems.
4. Den Defekt in die Dokumentation mit der Defektnummer zu markieren.

Die zukünftigen Inspektoren werden gebeten nach und während der Inspektion für jedes Dokument die Zeit anzugeben, die nötig war, um diese zu inspizieren und den Evaluierungsfragebogen bei jeder Inspektionsmethode auszufüllen.

Nachdem alle Aspekte einer Inspektionssitzung besprochen wurden, wurde ein Beispiel eines Fehlers des Anforderungsdokumentes der „Automatic Teller Machine“ angegeben. Es wurde gezeigt wie ein Defekt zu markieren ist und einer Defektklasse zugeordnet wird. Dazu dienten zwei Fehlerarten:

Weglassung der Zustandbeschreibung beim Initialzustand in den funktionalen Anforderungen, also Fehlerklassentyp „Omission“

Die Klasse „Miscellaneous“ wurde anhand einer unter Performanzanforderungen ausgeführte Anforderung erklärt.

Tatsächlich war dies aber eine funktionale Anforderung. (siehe Anhang)

Bevor wir zu den einzelnen Inspektionsverfahren gingen, haben wir erst mal die Inspektion definiert. Es wurde eine prägnante Zusammenfassung angegeben, nämlich dass Inspektionen wohldefinierte und disziplinierte Prozesse sind, die zu jeder Softwareentwicklungsphase durchgeführt werden können. Dazu dienen die verschiedensten Techniken.

In der Einführung haben wir insgesamt sechs Inspektionsmethoden vorgestellt, obwohl sie in der zweiten Stunde dann drei durchführen werden, so dass die Studenten einen breiteren Überblick über Inspektionen erhalten. Wir haben folgende Inspektionsmethoden kurz behandelt:

- Audit
- Review
- Walkthrough

Ausführlicher wurden dann die letztendlich zu benutzenden Inspektionsverfahren behandelt, in dem die Studenten in der Einführung einen Crash-Kurs über die Konzepte und der Durchführung der drei ausgewählten Prüfmethode erhalten haben. Diese Methoden sind:

- Die Ad-hoc Inspektion
- Die Checklisten basierte Inspektion(CLR)
- Die Perspektiven basierte Lesetechnik(PBR)

Diese drei Prüfverfahren wurden im Anschluss an die Einführung einzeln behandelt und extra anhand von Beispielen erläutert.

4.4.2 Ad-hoc

Die wesentlichen Ideen, die die Studenten beherzigen sollten bei der Ad-hoc Methode sind:

- Der Sinn auch dieser intuitiven Methode liegt darin, dass mehrere Leute immer mehr Fehler rausfinden als nur einer oder zu wenige. Auch wenn diese Technik eine geringe Überdeckung und eine hohe Überlappung der gefundenen Fehler aufweist(siehe Kapitel 2).
- Dass wir versuchen wollen, die Qualitätsanforderungen eines Wissensdokumentes zu ermitteln, um zu erreichen, dass weitere Benutzer keine Fehler in dem geprüften Dokument finden.

Beispiele von Fragen, die sich den Studenten beim Inspizieren von Wissensdokumenten laut der Ad-hoc Methode stellen könnten, sind:

- Kann ich die enthaltenen Informationen sinnvoll verwenden oder sind sie nützlich für meine Arbeit ohne weitere Erklärung?
- Wurden unnötige Informationen geführt?
- Wurden im Gegenteil wichtige Informationen weggelassen?

Wichtig ist auch immer auf die Konsistenz, Unklarheit oder sogar Inkorrektheit der enthaltenen Informationen genau zu achten.

4.4.3 Checklisten basierte Lesetechnik(CBR)

Da wir bei der CBR-Methode doch noch mehr Unterstützung haben in Form einer Liste von Fragen, können wir hier mehr auf die Aspekte eines Wissensdokumentes eingehen. Deswegen können wir uns spezielle Fragen zu jedem Aspekt stellen. Diese Fragen sollen das Wie, Warum und Wann ein Aspekt qualitativ gut genug ist, beantworten, und nicht aus wessen Sicht(Perspektive) dies der Fall ist. Das wäre dann die Aufgabe von der PBR-Lesetechnik.

Die Aspekte, die bei der CBR-Methode zu beachten sind, die am Anfang dieses Kapitels definierten Qualitätsmerkmale eines Wissensdokumentes. Dazu kommt noch die Frage nach dem Detaillierungsgrad, d.h. in wie weit bilden die Informationen eine adäquate Basis für die Anpassung der Lösung oder für die Anwendung des Wissensdokumentes.

Die Studenten kriegen eine Liste mit den Fragen, die sie während der Inspektionssitzung beantworten sollen, die sich natürlich an den oben vorgeführten Qualitätsmerkmale orientiert.

Wie bei der Ad-hoc Methode sollen die Fehler den Fehlerklassen zugeordnet werden.

Wie bereits erwähnt, sind die Fragen nach Syntax und Semantik unterteilt. Da ein Wissensdokument wie wir alle mittlerweile beherzigt haben, meistens nicht klar genug ist, nicht strukturiert und zu allgemein ist, sollte diese Unterteilung den Studenten eine willkommene Hilfe sein.

Hier sind ein paar Fragen aus der Checkliste zu jedem Qualitätsmerkmal. Die komplette Checkliste finden Sie im Anhang dieser Diplomarbeit.

- Klarheit: Sind die Aussagen in dem Wissensdokument deutlich?
- Vollständigkeit: Sind das Problem, die Lösung, der Kontext, die Ziele, der Zweck, die Annahmen, die Beschränkungen und der Erfahrungsbereich vollständig in dem Wissensdokument aufgelistet?
- Konsistenz: sind die Aussagen im Wissensdokument alle untereinander konsistent. D.h. widerspricht eine Aussage keiner anderen innerhalb des Dokumentes?
- Korrektheit: beschreiben die Aussagen das Problem, die Lösung, etc auf einer korrekten Weise?

- Funktionalität: sind alle beschriebenen Schritte der Lösung/Aussagen notwendig und ausreichend für die Zwecke des Wissensdokumentes?
- Testbarkeit: kann das Wissensdokument alleine getestet werden ohne größeren Kontext, z.B. innerhalb eines realen Projektes
- Detaillierungsgrad: bieten die Aussagen eine adäquate Basis für die Anwendung oder Anpassung der Lösung?

Durch die Unterstützung der Checklisten werden bessere Ergebnisse als bei der Ad-hoc Methode erwünscht. Dies wird sich bei der Evaluation dann zeigen

4.4.4 Perspektiven basierte Lesetechnik

Festzuhalten bei dieser Lesetechnik ist, dass ihr die sogenannte Stakeholderanalyse zugrunde liegt, wie im Abschnitt über die Entwicklung der Methode beschrieben ist. Daher müssen wir verschiedene Interessensträger definieren und die Qualitätsmerkmale eines Wissensdokumentes aus der Perspektive des einzelnen Stakeholders versuchen zu genügen.

Dadurch dass die Gutachter in die Lage einer bestimmten Perspektive versetzt werden, wird auch eine gewisse Unabhängigkeit von externen Faktoren, wie die persönliche Erfahrung des einzelnen Gutachters, mit dem selben Projekt oder mit anderen Projekten gewährleistet. Die Ausführung der angegebenen Aktivitäten hat zum Ziel, das Lenken der Gutachteraufmerksamkeit auf bestimmte Aspekte des Dokumentes, wodurch eine garantierte Hilfe beim Verständnis des Dokumentes erhofft wird. Dieses Verständnis bildet wiederum die Grundlage für eine bessere Dokumentenanalyse und somit unterstützt es den Gutachter beim Finden von Fehlern. Zusätzlich zu diesen speziellen Fehlern, können die Inspektoren natürlich auch allgemeine oder andere Fehler finden, die nicht aus dem Szenario hervorgehen, sondern aus der persönlichen Erfahrung oder aus sonstigen allgemeinen Kenntnissen. Sie werden dann auch dazu angehalten, diese Fehler gesondert zu dokumentieren.

Um sicherzustellen, dass wir die verschiedenen Perspektiven bedienen, gehen wir erst mal der Frage nach, wer die Interessensträger eines Wissensdokumentes überhaupt sein können? Und wie könnten ihre Interessenschwerpunkte aussehen?

Ganz intuitiv kommt man als allererstes auf den **Benutzer** des Wissensdokumentes. Ein weiterer Interessensträger ist der **Autor** dieses Wissensdokumentes. Der **Leser** stellt wieder andere Ansprüche an das Wissensdokument. Weitere Perspektiven sind der **Qualitätsmanager** und der **Tester**. Man kann mit der Definition fortfahren, aber wichtiger ist hier Beantwortung der Frage, was die gemeinsamen Perspektiven der verschiedensten Wissensdokumente sein könnten. Für den Zeitrahmen dieses Experiment müssen nicht mehr als drei Perspektiven definiert werden. Die Auswahl fiel auf den Autor, den Leser und den Tester eines Wissensdokumentes. Diese drei Perspektiven sind aus der Sicht dieser Diplomarbeit für diese Studie sehr gut geeignet, da sie zum einen allgemeingültig für die meisten Wissensdokumente sind, und zum anderen von den Studenten leicht nachvollziehbar sein können. Denn jedes Dokument beansprucht intuitiv einen Autor, und einen Leser. Für das Thema Inspektion ist die Perspektive Tester gut geeignet und erforderlich.

Nachdem die Interessensträger standen, wurde dann ein Rahmen definiert, in dem die Fragen gestellt werden können. Diese Fragen sollten genauso wie bei der Checklisten basierten Lesetechnik differenziert betrachtet werden in Anlehnung an die am Anfang dieses Kapitels definierten Qualitätsmerkmale eines Wissensdokumentes. Um diesen Qualitätsmerkmalen zu genügen, wurden die Fragen nach vier Kategorien klassifiziert:

1. Die syntaktische Fehlerklasse
2. Die semantische Fehlerklasse
3. Die kontextuelle Fehlerklasse
4. Eine Kategorie für Verschiedenes, die zur freien Verfügung stellt für alle möglichen Aspekte.

In Anlehnung an diese Fehlerklassen werden je nach Perspektive Fragen definiert. Hier folgen ein paar Beispielfragen für die einzelnen Perspektiven.

4.4.4.1 Autor

Die Aufgabe eines Autors ist das Wissen und den Kontext klar und strukturiert niederzuschreiben. Folgende Fragen würden einem Inspektor helfen, dieses Ziel auf seinen Erfüllungsgrad zu überprüfen.

Die syntaktischen Fragen sind z.B.:

Ist das Dokument gut strukturiert?

Wurden wichtige und zum Verständnis des Dokumentes notwendige Informationen weggelassen?

treten in dem Dokument Redundanzen auf?

Die semantischen Fragestellungen könnten sein:

Ist das Dokument gut lesbar für Sie als Experte?

Kann das Dokument im selben Kontext anders interpretiert werden von anderen Lesern?

Eine kontextuelle Frage könnte lauten:

Reicht die angegebene Zeit zum Lesen und Verstehen des Dokumentes?

Auch hier werden die Fehler nach den vorgestellten Fehlerklassen klassifiziert.

4.4.4.2 Der Tester

Das wesentliche Ziel beim Testen ist das Wissensdokument in seinem Anwendungsbereich auf die Ausführbarkeit hin zu überprüfen.

Die Fragen würden dann beispielsweise lauten je nach Kategorie:

Syntaktisch gesehen:

Ist das Dokument vollständig?

Erfüllt das Dokument die geforderten Standards?

Auf der semantischen Ebene:

Erfüllt das Dokument die geforderten funktionellen Standards?

Konnte das angegebene Problem mit der angegebenen Lösung gelöst werden?

Gibt es Ihrer Meinung nach eine andere alternative Lösung? D.h. ist die Lösung zu allgemein?

Ist die Lösung Ihrer Meinung nach auch für andere Probleme geeignet? D.h. wie speziell ist die Lösung?

4.4.4.3 Leser

Die Hauptaufgabe des Lesers ist sicherzustellen, dass das Wissensdokument verständlich, klar und konsistent ist und dass das darin enthaltene Wissen in einem anderen Kontext gut anwendbar ist.

Um dies sicherzustellen, können folgende Frage hilfreich sein:

Auf der syntaktischen Ebene:

Ist das Dokument gut strukturiert?

Werden wichtige und zum Verständnis des Dokumentes notwendige Informationen vermisst?

treten in dem Dokument Redundanzen auf?

Die semantische Fehlerklasse könnte wie folgt vertreten sein:

Ist das Dokument gut lesbar?

Brauchen Sie extra Hilfe zum Verständnis des Dokumentes?

Wurde Ihnen die zum Verständnis des Dokumentes notwendige Hilfe zur Verfügung gestellt?

Kann das Dokument im selben Kontext anders interpretiert werden von anderen Lesern?

Eine kontextuelle Frage könnte lauten:

Reicht die angegebene Zeit zum Lesen und Verstehen des Dokumentes?

4.5 Die Ausführung des Experimentes

Im zweiten Abschnitt des Experimentes wurden unsere potentiellen Inspektoren an die Arbeit geschickt.

Die Studenten kriegen vor jeder Inspektion folgende Formulare:

- Eintragungsformular
- Die Checkliste zu der CBR Inspektion, an die die Studenten sich orientieren sollen
- Prozessbeschreibung
- Ein Evaluierungsformular zu den einzelnen Wissensdokumenten
- Ein Evaluierungsformular zu den Inspektionsmethoden
- Das zu inspizierende Wissensdokument

Sie müssen dann die ausgefüllten Formulare zurückgeben, die Kommentare und die Evaluierungsformulare der einzelnen Prüftechniken.

4.5.1 Die Wissensdokumente

Bevor die Dokumente vorgestellt werden, die die Studenten dann tatsächlich am Ende erhielten, sei noch erwähnt, dass wir vorhatten

ein paar Defektfälle von dem Defekt Management System, Bugzilla von unseren Studenten inspizieren zu lassen. Leider hat das nicht stattgefunden, da die meisten Defektberichte nach sorgfältiger Überprüfung sich überhaupt nicht dazu geeignet erschienen. Das Problem dabei war die fehlende ausführliche Beschreibung der endgültigen Lösung. So hat man keine Möglichkeit gehabt bei den Defekt/Lösungspaaren die Lösungen auf ihre Tauglichkeit zu überprüfen.

Als externe Dokumente dienten Dokumente aus dem Virtuellen Software Engineering Kompetenzzentrum (Visek), das vom Bundesministerium für Bildung und Forschung gefördert wird. Das IESE ist eine von den acht Forschungseinrichtungen, die das Projekt aufgebaut haben. In diesem Portal sind viele Wissens- und Erfahrungsdokumente vom Bereich Software Engineering zur freien Verfügung gestellt. (siehe www.visek.de)

Die internen Dokumente des IESE kamen aus verschiedenen Experience Base Demonstratoren. Diese sind Teil des CoIN (Corporate Information Network)- Projektes am IESE, das sich mit Erfahrungsmanagement befasst und eine Experience Factory zur Verfügung stellt [BCR94b]. Die Factory beinhaltet folgendes:

1. Prozessbeschreibung zu den Aktivitäten innerhalb des CoIN Projektes
2. Rollenbeschreibungen zu den Rollen innerhalb der Prozesse
3. Agentenbeschreibungen, die innerhalb der Rollen benutzt werden
4. Produktbeschreibungen, die Dokumente beinhaltet, die während der Durchführung eines Prozesses benutzt werden können
5. Übersichtsstrukturen, die das Navigieren innerhalb des IQ erleichtern

Nachdem die beiden Ressourcen vorgestellt wurden, werden nun die ausgewählten Dokumente vorgestellt:

Aus Visek wurden folgende Dokumente ausgesucht:

- Produktlinien: ein Einstieg.
- Testmanagement
- Aus dem Experience Base Demonstrator Dokumente zu:
- Konfiguration von Eclipse für Tomcat-Projekte
- Html-Formulare: Erkennung von nicht gesetzter Checkbox
- Auszüge aus der Projekterfahrungsdatenbank
- Prozessbeschreibung mit der case_id2860
- Prozessbeschreibung mit der case_id2435

Dazu haben die Studenten noch Prozessbeschreibungen aus dem IESE Informationsbestand erhalten, diese sind:

- Conference Participation Planning
- Thesis Work

4.5.2 Der Ablauf des Experimentes

Die Studenten fingen mit der Ad-hoc Methode an, da diese sich als Aufwärmübung bestens geeignet ist. Sie erfordert keine Einarbeitung, sodass sie sich mit der Inspektion an sich vertrauter fühlen können. Die Studenten können mit dem Lesen der Wissensdokumente anfangen, und bei jedem gefundenen Fehler diesen markieren und in das dafür vorgesehene Formular eintragen.

Die Studenten sollten laut dem ursprünglichen Plan mit der Checklisten basierten Lesetechnik weitermachen. Da es mehr als 50 Minuten für die Ad-hoc Methode von den Studenten gebraucht wurden, mussten die Entscheidung für die restlichen Minuten für eine der beiden Lesetechniken fallen. Die Entscheidung ist nicht schwergefallen. Da die Perspektiven basierte Lesetechnik aufgrund

der Erkenntnisse aus der Softwareinspektionsstudien effektiver ist, und außerdem die Erarbeitung der einzelnen Szenarien, deren Anweisungen und Fragen für die einzelnen Perspektiven mehr Zeit und Aufwand erfordert hatte als die Erarbeitung von den Fragen zu der Checklisten basierten Lesetechnik, wurde direkt mit der Perspektiven basierten Lesetechnik weitergemacht.

4.6 Evaluation

In diesem Abschnitt sollen die Erkenntnisse, die aus der Evaluierung des Inspektionsexperimentes gesammelt wurden, vorgestellt.

Nachdem wir die ausgefüllten Formulare der Studenten und deren Kommentare erhalten hatten, mussten wir nun überprüfen, inwiefern die angegebenen Fehler auch tatsächlich solche sind. Die Fehler wurden laut Tabelle 1 auf Seite 73, die vorher in der Einführung zum Abschnitt Experiment vorgestellt wurde, klassifiziert.

Bei der Evaluierung haben wir folgende Erkenntnisse gesammelt:

- Die meisten gefundenen Fehler sind vom Typ Omission. Es wurden 18 Fehler dieses Typus gefunden, die wir tatsächlich auch bei der Überprüfung als solche gelten ließen. Bei diesem Fehlertyp handelt es sich wie der Name schon sagt um fehlende Informationen über das Problem bzw. die Lösung oder über den Kontext, die zum Gesamtverständnis des Dokumentes ununterlässlich sind.
- Dicht daran wurde die Fehlerklasse Ambiguous Information mit 11 gefundenen Fehlern vertreten. Unter dieser Fehlerklasse lassen sich alle unklaren und mehrdeutigen Informationen innerhalb des Wissensdokumentes einordnen. Diese Information ist so unpräzise, dass der Benutzer bzw. Leser des Dokumentes sich dazu verführt fühlt, subjektive Interpretationen in das Dokument einfließen zu lassen.
- Es wurde nur einmal von unseren Inspektoren einen Fehler der Klasse Extranous gefunden. Dieser Fehlertyp besagt, dass die gegebene Information redundant ist oder gar nicht benutzt

wird. Auf jeden Fall handelt es sich um eine überflüssige Information.

- Die restlichen acht Fehler, die noch von unseren Inspektoren angegeben waren, wurden der Fehlerklasse Miscellaneous zugeordnet. Unter dieser Fehlerklasse lassen sich alle Fehler unterordnen, die sich keiner der wohldefinierten Fehlerklassen zuordnen lassen. Es sind die verschiedensten Fehler, wie Klein- und Großschreibung, Grammatikfehler und ähnliches.

Folgendes Diagramm hält diese Ergebnisse bildlich fest:

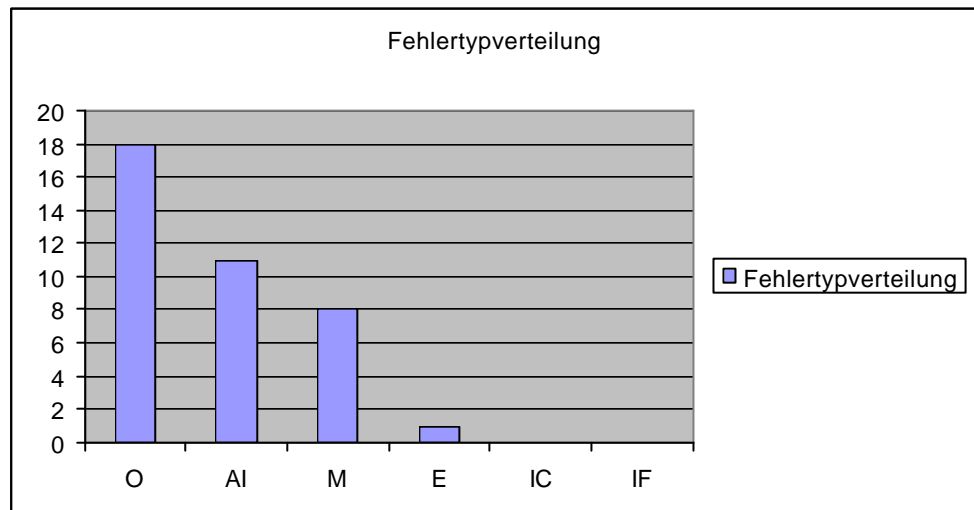


Diagramm 1

Gesamte Verteilung der Fehlerklassen

Die Evaluation dieser Inspektion hat Fehler ans Tageslicht gebracht, die auf jeden Fall korrigiert werden müssen. Diese sind entweder semantischer oder syntaktischer Art. Die semantischen Fehler reichten von fehlenden Erklärungen zu den Abkürzungen bis hin zu ganz fehlenden Begriffsdefinitionen. Die syntaktischen Fehler waren zum Teil auch falsche Aufbaustruktur des Wissensdokuments, wie auch einfache Grammatikfehler.

Nun wollen wir uns die Frage stellen, welche Methode wie viele Fehler aufzuweisen hat.

4.6.1 Einzelbewertung der Ad-hoc Technik

Bei der Ad-hoc Inspektionsmethode betrug die Anzahl der insgesamt gefundenen Fehler 29, davon waren 14 vom Typ Omission(O), 11 vom Typ Ambiguous Informatio(AI) und 4 vom Typ Miscellaneous (M), siehe folgendes Diagramm.

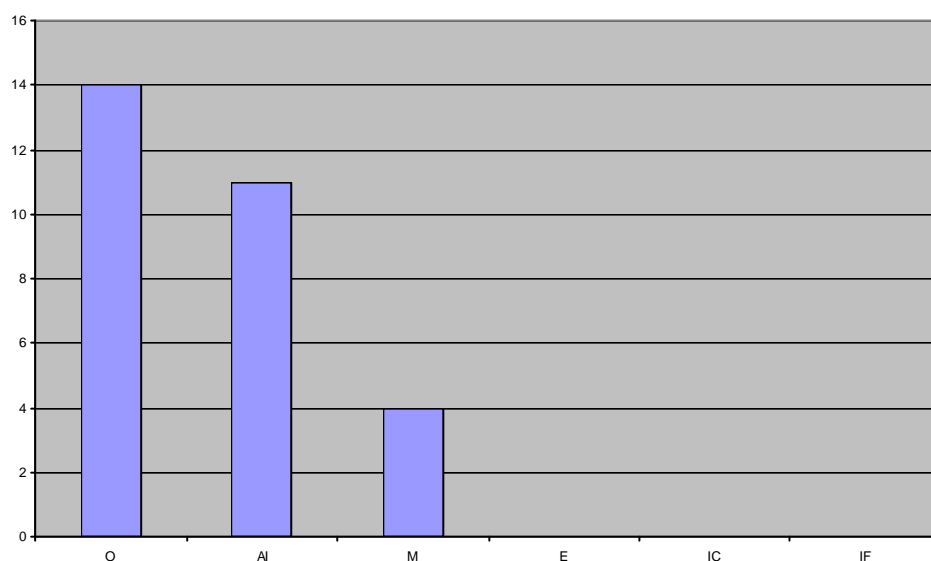


Diagramm 2 Fehlerverteilung bei Ad-hoc

Die Dokumente, die allen drei Inspektoren für die Ad-hoc Methode ausgeteilt wurden, waren:

1. Aus dem Experience Base Demonstrator das Problem-Lösungspaar zu: Konfiguration von Eclipse für Tomcat
2. Aus der Projektdatenbank des IESE den Fall mit der Nummer 2860 vom 05.08.2002
3. Conference Participation Planning aus der Prozessdatenbank des IESE
4. Von Visek haben wir das Prozessmuster: Produktlinien - Einstieg

Alle Dokumente finden Sie im Anhang dieser Diplomarbeit.

Als nächstes möchten wir die Verteilung der gefundenen Fehler auf die einzelnen Inspektoren uns anschauen.

Es sei zuerst darauf hingewiesen, dass manchen Inspektoren aus Zeitgründen nicht die ganzen Dokumente inspizieren konnten, dies ist mit „nicht ermittelt“ angegeben.

Anzahl der gefundenen Fehler von	Inspektor A	Inspektor B	Inspektor C	Insgesamt
CPP	4	5	5	8+1neu
Produktlinien	1	1	3	3
EBD	5	2	nicht ermittelt	6
PE2860	1	1	1	2

Tabelle 2

Verteilung der Fehler im Verhältnis zu den Inspektoren

Folgendes Diagramm gibt uns noch mal einen Überblick:

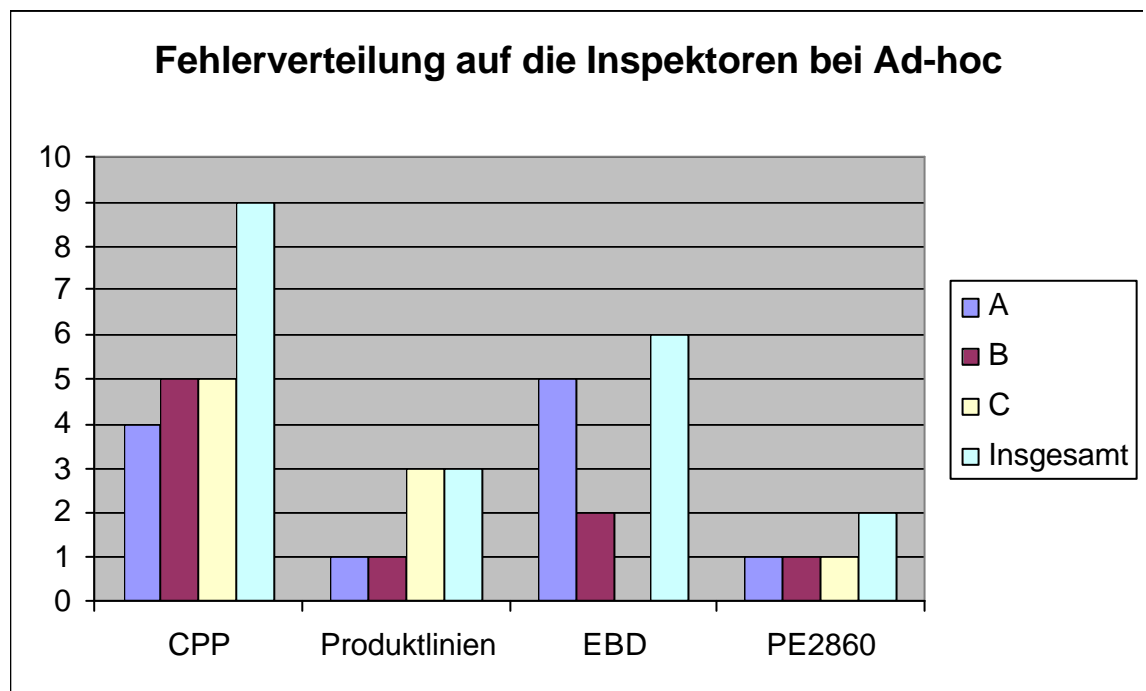


Diagramm 3

Fehlerverteilung auf die einzelnen Inspektoren und insgesamt

4.6.2 Einzelbewertung der PBR-Methode

Bei der Perspektiven basierten Lesetechnik:

1. Aus dem Experience Base Demonstrator das Problem-Lösungspaar zu HTML-Formularen: Erkennung von nicht gesetzter Checkbox
2. Aus der Projektdatenbank des IESE den Fall mit der Nummer 2435 vom 18.08.2000
3. Thesis Work aus der Prozessdatenbank des IESE
4. Von Visek haben wir die Methodenbeschreibung zu Testmanagement

Auch diese Dokumente sind im Anhang zu finden.

Bei der PBR Lesetechnik wurden insgesamt 18 Fehler gefunden, davon 13 vom Typ Omission(O), 5 vom Typ Miscellaneous(M) und 2 vom Typ Ambiguous Information(AI), wie folgendes Diagramm zeigt.

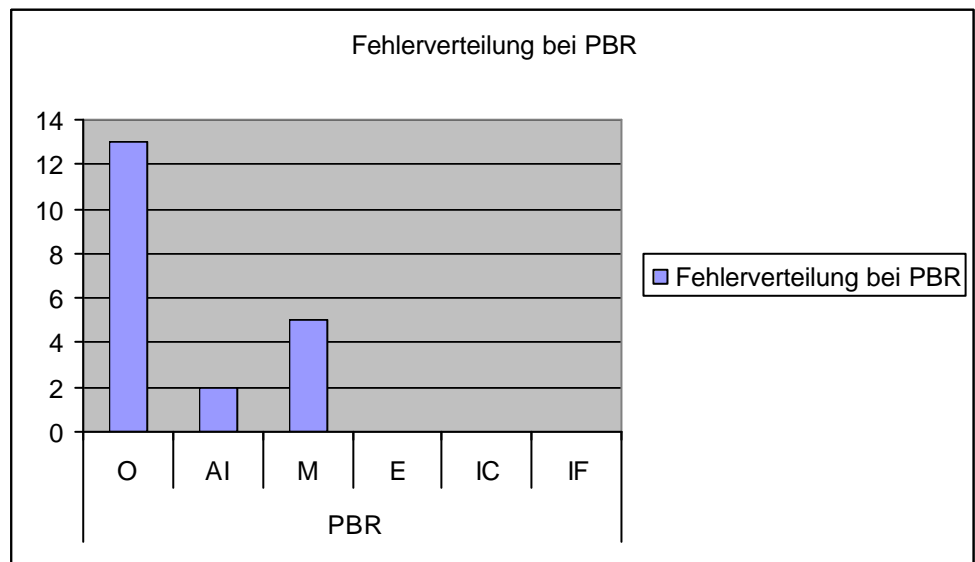


Diagramm 4

Fehlerklassenverteilung bei PBR

Nun möchte ich der Frage nachgehen, welche Fehler bei welchen Dokumenten gefunden wurden. Die Antwort steht in der folgenden Tabelle:

Anzahl der gefundenen Fehler von	Inspektor A	Inspektor B	Inspektor C	Insgesamt
TW	6	2	4	10
Testmanagement	Nicht ermittelt	Nicht ermittelt	1	1
SE135	4	Nicht ermittelt	1	5
PE2435	1	1	Nicht ermittelt	2

Tabelle 3

Verteilung der Fehler im Verhältnis zu den Inspektoren

Auch hier das Digramm zu der Fehlerverteilung auf die einzelnen Inspektoren und insgesamt:

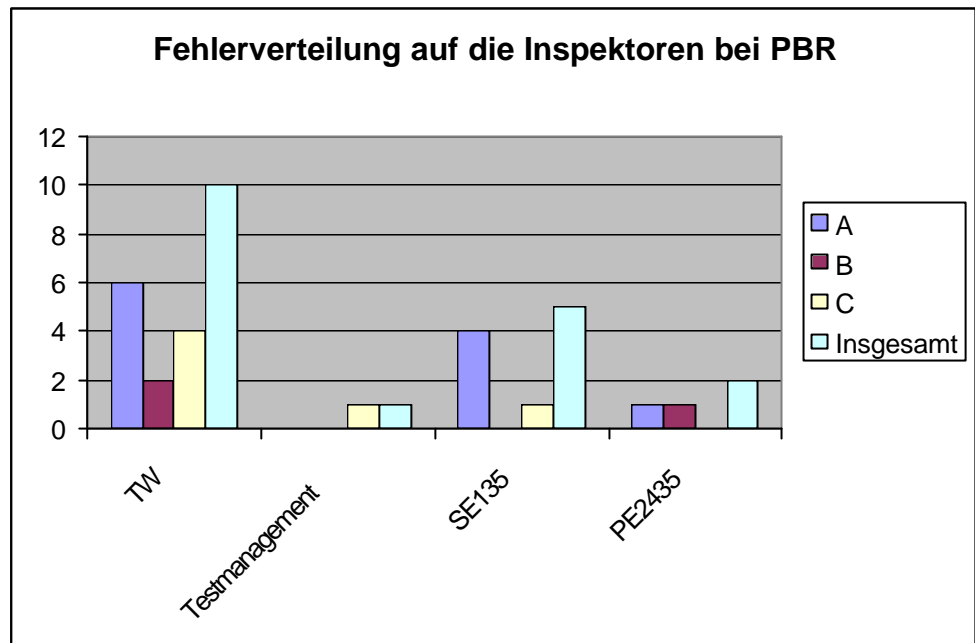


Diagramm 4

Fehlerverteilung bei PBR

4.6.3 Vergleich der Methoden Ad-hoc und PBR

Bei beiden Inspektionsmethoden ist auf jeden Fall folgendes festzuhalten:

Die Gesamtanzahl der gefundenen Fehler von den Inspektoren, wenn diese Fehler gefunden haben, ist stets größer als die Anzahl der gefundenen Fehler von einem einzelnen Inspektor. Dies beweist auf jeden Fall die These, dass es immer sinnvoll ist, mehrere Inspektoren an einer Inspektion teilnehmen zu lassen.

Zu den jeweiligen Inspektionsmethoden können wir auf jeden Fall sagen, dass während bei der Ad-hoc Methode die Fehler sich mehr allgemein halten, in Form von fehlenden bzw. mehrdeutigen Informationen, die Fehler bei der Perspektiven basierten Lesetechnik sich mehr auf die Syntax bzw. Die Semantik der Dokumente. Dies ist Mithilfe der Anleitung, die die Inspektoren erhalten hatten, erreicht

worden. Die entwickelten Fragen für jede Perspektive haben die Aufmerksamkeit der Inspektoren auf jeden Fall auf die wichtigen Merkmale jedes Dokumentes fokussiert. Die Zeit, die wir bei der Entwicklung dieser Methode investiert haben, hat sich auf jeden Fall rentiert, da zum Einen erheblich weniger Zeit bei der Perspektiven basierten Lesetechnik von unseren Inspektoren gebraucht wurde, egal wie lang das Wissensdokument war. Und zum anderen detailliertere Angaben zu den Fehlern gemacht worden sind.

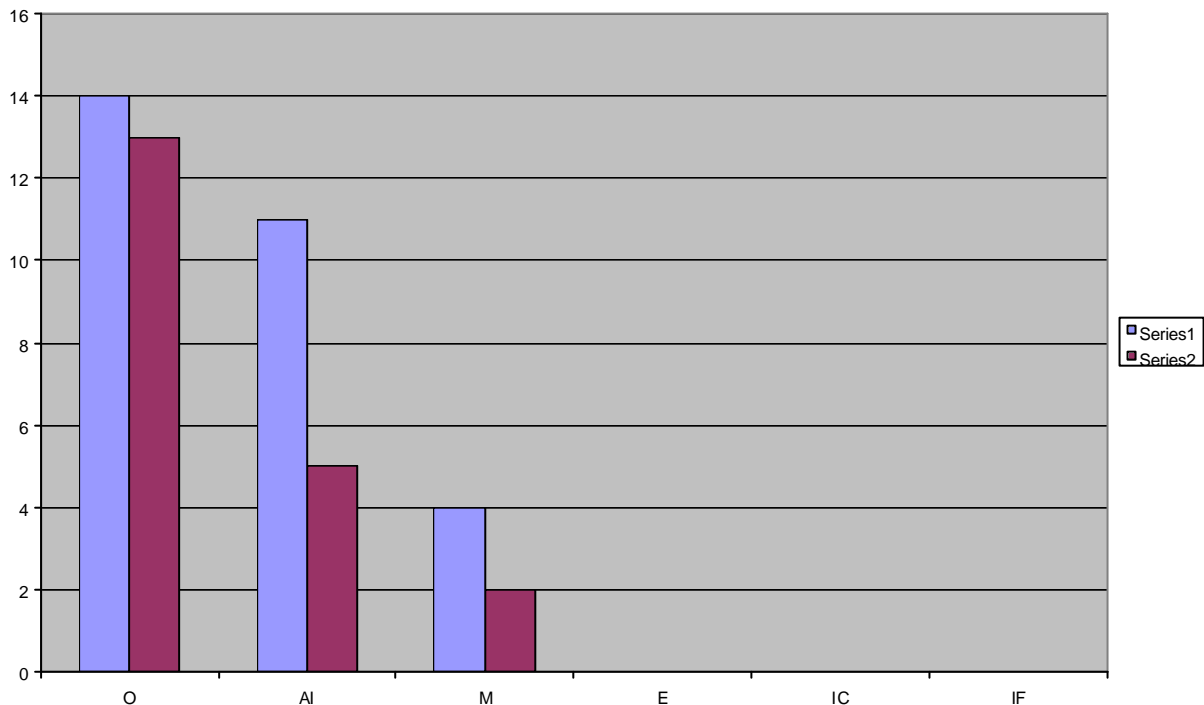


Diagramm 5 Ad-hoc und PBR im Vergleich

4.7 Zusammenfassung

Zusammenfassend lässt sich sagen, dass wir auch auf dem Feld der Wissensinspektion, viele Erkenntnisse aus den Experimenten mit der Softwareinspektion bestätigen können. Jede Inspektion hat ihre Vor- und Nachteile. Diese lassen sich zusammenfassen so sagen:

4.7.1 Ad-hoc:

Vorteile:
<ul style="list-style-type: none"> • Wenig Anpassungsaufwand auf die Wissensdokumente • Intuitiv • Überall anwendbar • Wenig Einarbeitungsaufwand für die Inspektoren
Nachteile:
<ul style="list-style-type: none"> • Bietet keinerlei Unterstützung • Hängt stark vom individuellen Wissen und Erfahrungen der einzelnen Inspektoren • Braucht längere Sitzungen • Hohe Überlappung • Niedrige Überdeckung

Tabelle 4

Vor- und Nachteile bei der Ad-hoc Inspektion

4.7.2 Checklisten basierte Lesetechnik:

Vorteile:
<ul style="list-style-type: none"> • Mittlerer Anpassungsaufwand auf die Wissensdokumente • Mehr Unterstützung durch Checklisten • Überall anwendbar • Wenig Einarbeitungsaufwand für die Inspektoren
Nachteile:
<ul style="list-style-type: none"> • Konnten leider nicht festgestellt werden, da diese Technik beim

Experiment nicht durchgeführt werden konnte.

Tabelle 5

Vor- und Nachteile bei der CLR Inspektion

4.7.3 Perspektiven basierte Lesetechnik:

Vorteile:
<ul style="list-style-type: none">• Beste Unterstützung durch Szenarien und Checklisten• Hängt nicht mehr so stark vom individuellen Wissen und Erfahrungen der einzelnen Inspektoren• Braucht kürzere Sitzungen• Überall anwendbar• Hohe Überdeckung• Niedrige Überlappung
Nachteile:
<ul style="list-style-type: none">• Hoher Anpassungsaufwand auf die Wissensdokumente, Ableitung der Rollen und Szenarien braucht viel Zeit• Braucht mehr Einarbeitungszeit für die Inspektoren

Tabelle 6

Vor- und Nachteile bei der PBR Inspektion

Dies lässt uns folgende Behauptung stellen:

Gerade bei Wissensdokumenten, die meistens nicht leicht verständlich sind, kontextabhängig und nicht strukturiert sind, haben sich die speziell zu den jeweiligen Perspektiven definierten Checklisten als sehr hilfreich erwiesen. Die Inspektoren haben sich besser zurecht gefunden, und wussten wonach sie Ausschau halten sollten. Dies war möglich, auch wenn die Inspektoren keine ausführliche Einarbeitung genossen hatten, und über wenig bzw. keine Fachkenntnisse verfügten. Auch die gefundenen Fehler waren Dank der Checklisten nicht mehr allgemein formuliert. Dies bedeutet,

dass der Aufwand, der für die Ableitung der Szenarien und Checklisten auf jeden Fall sich gelohnt hat.

Nicht desto Trotz wurden durch das Ad-hoc Vorgehen auch Fehler entdeckt. Dies besagt, dass auf jeden Fall die Durchführung einer Inspektion sinnvoll und ratsam ist. Jeder müsste dann nur noch für seine eigenen Bedürfnisse die Vor- und Nachteile jeder Inspektion abwägen.

5 Zusammenfassung und Ausblick

In dieser Arbeit haben wir versucht, ein Stück näher in Richtung wissenschaftlich fundierte Erkenntnisse über Wissensinspektion zu kommen. Dies ist heutzutage genauso notwendig, wie es bei Software Inspektionen vor ein paar Jahrzehnten notwendig war. Auch da mussten etliche wissenschaftliche Untersuchungen auf den verschiedensten Ebenen vorliegen, und enorme Überzeugungsarbeit geleistet werden, ehe sich die Softwareinspektion soweit, glaubhaft und fundiert in vielen Softwareentwicklungsfirmen etabliert hat.

In unserer heutigen Zeit geht nun der Trend in Richtung wissensorientierte Organisation. Dies ist so im Aufbruch, dass viele Ideen und Strategien geradezu aus den Köpfen vieler ehrgeizigen Wissenschaftlern herauspringen. Angefangen von der Frage wie man Wissen definieren soll, über die Frage wie man Wissen organisieren soll, bis hin zu der Frage, was eigentlich die Wissenskomponenten sind. Die Antworten zu diesen Fragen findet man in vielen Büchern, die sich alle mit Wissen und Wissensmanagement in Organisationen beschäftigen.

5.1 Ziele der Arbeit

Diese Diplomarbeit handelt nicht nur über Wissensmanagement. Vielmehr wollen wir hiermit auch ein wissenschaftlicher Grundstein zu Wissensinspektion anlegen in Anlehnung an die heute so erfolgreich praktizierte Software Inspektionen.

Zu diesem Zweck wurden folgende Ziele festgesetzt:

1. Recherche über die theoretischen Grundlagen von Inspektionen im Software Engineering, über Wissen und Wissensmanagement.
2. Auswahl von drei Methoden aus den wohldefinierten Inspektionsmethoden im Software Engineering, die sich zum Inspizieren von Wissensselementen eignen werden.

3. Verfeinerung und Anpassung der ausgewählten Prüfverfahren zu den Merkmalen und Eigenschaften von Wissensselementen, unter Berücksichtigung der Problembereiche von manchen Wissensseigenschaften.
4. Experiment und Evaluierung, bei dem die Prüfverfahren durchgeführt, analysiert und evaluiert wurden.

Das Kapitel zwei diente der Recherche über die meisten heute verfügbaren Software Inspektionsmethoden und war ein wichtiger Bestandteil dieser Arbeit. Ich habe versucht, gründliche Recherchen auf diesem Gebiet zu betreiben, und mich an den Pionieren der Software Inspektionsentwicklung orientiert, um dann ihre entwickelten Methoden vorzustellen. Dabei kann ich zusammenfassend sagen, dass sich die Ad-hoc, Checklisten basierte und Perspektiven basierte Lesetechniken in der Gemeinschaft als allgemein anerkannte Inspektionsmethoden etabliert haben. Wobei die Unterschiede sich im wesentlichen nur auf Aufwand/Effizienz Verhältnis beschränken. Beispielsweise ist die Ad-hoc Inspektion einfach, schnell und überall einsetzbar. Sie bietet dafür dem Inspektionsteam keinerlei Unterstützung, so dass die dabei erzielte Effizienz nicht so hoch ist, und der Grad an Überlappung hoch ist während die Überdeckung niedrig gehalten wird. Dagegen ist die Perspektiven basierte Lesetechnik eine wohldefinierte Methode, die Szenarien für jede Rolle vorsieht, so dass wenig Raum den Inspektoren gegeben wird, die persönliche Erfahrung in das Dokument einfließen. Dadurch wird gewährleistet, dass die Dokumente objektiv und nur nach den vorgesehenen Szenarien überprüft werden. Der hohe Aufwand, der bei der Entwicklung der Szenarien, der darin enthaltenen Aktivitäten und Checklisten entsteht schreckt manche ab. Allerdings ist die dabei erzielte Effizienz auch sehr hoch, was nicht zuletzt auf die hohe Überdeckung und niedrige Überlappung zurückzuführen ist.

Nachdem der Überblick über die verschiedensten Inspektions-Methoden feststand, wurde mit der Recherche über Wissen und Wissensmanagement weitergemacht. Dies war Thema des dritten Kapitels. Erst wurden die verschiedenen Definitionen von Wissen behandelt. Dann wurde der Anreicherungsprozess vorgestellt, der die

Grenzübergänge zwischen Daten, Information und Wissen genau festlegt. Darüber hinaus wurde das Wissen aus der Epistemologischen Sicht betrachtet, das sich mit Metawissen, latentem Wissen, explizitem und implizitem Wissen beschäftigt. Die letzten beiden Formen sind besonders von Wichtigkeit, da die sich beide ineinander umformen lassen. Dazu haben wir das Spiralenmodell von Nonaka vorgestellt [Non97].

In der Realität tritt aber Wissen nicht in einer reinen Form von explizitem oder implizitem Wissen. Daher habe ich es für wichtig gehalten, das HANSE-Modell vorzustellen, das sich genau mit den Anteilen von jeder Wissensform an **H**euristiken, **A**rtefakte, **N**atürlichen Begabung, **S**kills und **E**rfahrungen beschäftigt.

Dann wurden die Wissensträger vorgestellt bevor die Wissensdarstellungsformen eingeführt wurden und dann der Begriff Wissensmanagement eingeführt wurde. Hierzu wurden verschiedene Definitionen vorgestellt bevor ich mich auf die Konzepte von Probst [Pro97] eingeschränkt hatte. Hierzu sind die Bausteine des Wissensmanagements von Probst [Pro97] gründlich. Anschließend wurden die verschiedenen Arten von Wissensmanagement vorgestellt.

Zu dem Wissen und Wissensmanagement kann hier gesagt werden, dass diese Thema sich immer noch im Aufbruch befindet. Es wurden verschiedene Konzepte von verschiedenen Experten definiert. Es ist für jeden genug Auswahl, um das richtige Konzept für die speziellen Bedürfnisse der eigenen Firmen herauszufinden. Auf diesem Bereich wird immer noch daran gearbeitet, wie Wissen zu organisieren und zu managen ist. Literatur dazu findet man mehr als ausreichend, so dass ich irgendwann mit der Recherche auf diesem Gebiet abbrechen musste, damit diese Diplomarbeit zu ihrem wesentlichen Zweck noch kommen konnte.

Nun waren wir gut genug gerüstet mit theoretischen Hintergrundwissen auf der einen Seite zu den Inspektionsarten, die im Software Engineering zur Zeit existieren, und auf der anderen Seite mit genug Informationen über das Wissen an sich, seine

Darstellungsformen und Bausteinen. Der Weg war nun gut bepfastert zur Entwicklung unseres Experiment.

5.2 Experimentelle Beschreibung und Auswertung

Bei dem Experiment ging es zuerst in Anlehnung an die Inspektionsmethoden im Software Engineering, geeignete Methoden auszusuchen. Es war von Anfang an klar, dass man sich im Rahmen einer Diplomarbeit sich auf einige wenige Prüfverfahren einschränken sollte, um diesen dann die nötige Zeit und genug Entwicklungsraum bieten zu können.

Es hat sich herausgestellt, dass sich viele Inspektionsmethoden von vorneherein nicht eignen, da sie für spezielle Dokumente entwickelt worden sind wie z.B. das Lesen durch schrittweise Abstraktion, das speziell für Codefragmente benutzt wird (siehe Kapitel 2). Da aber unsere Wissensdokumente vielfältig sind, muss Ausschau gehalten werden nach Methoden, die sich für die meisten Dokumente eignen.

Die Entscheidung fiel wie in dieser Arbeit schon erklärt ist, auf diese drei Methoden entschieden:

1. Die Ad-hoc Methode, weil sie intuitiv ist und daher schnell einsetzbar ist. Also eignet sie sich auch für unerfahrene Inspektoren und für alle Dokumentarten. Sie liefert im Bereich von Software Engineering gute Ergebnisse. Dies erhoffen wir uns auch für unsere Erfahrungs- und Wissensdokumente.
2. Die Checklisten basierte Lesetechnik, da sie nicht viel schwieriger als die Ad-hoc Methode ist, sie aber durch die Fragenliste den Inspektoren eine bessere Unterstützung bietet. Dabei mussten wir bei der Entwicklung von den Fragelisten darauf achten, die Fragen nach Semantik und Syntax zu trennen. Dadurch wird zusätzlich zu den Checklisten den Inspektoren etwas mehr Unterstützung geboten, so dass die Wissensdokumente dadurch doch etwas mehr Struktur erhalten.

3. Gespannt wurde auf die Erfahrung und Ergebnissen von der Perspektiven basierten Lesetechnik gewartet, da diese doch die bessere Effektivität im Bereich Software Engineering erzielt, die bessere Unterstützung den Inspektoren bietet in Form von genau definierten Szenarien, Anweisungen und Fragelisten innerhalb der einzelnen Szenarien. Dabei wird erhofft, aus den verschiedenen Szenarien dann tatsächlich wie im Software Engineering eine gute Überdeckung und geringe Überlappung zu erzielen.

5.3 Lessons learned

Bei der Durchführung von dem Experiment haben wir dann festgestellt, dass angesichts der fehlenden Erfahrung der Studenten mit der Durchführung einer Inspektion, der geplante Zeitrahmen nicht passend war. Es musste dann letztendlich auf eine der beiden restlichen Inspektionen verzichtet. Die Checklisten basierte Lesetechnik musste dann im praktischen Teil ausfallen um dann mit der Perspektiven basierten Lesetechnik sofort nach der Ad-hoc Inspektion weiterzumachen. Denn der Aufwand zur Entwicklung der Perspektiven basierten Lesetechnik war erheblich höher.

Die Studenten hätten gerade mit den nicht mehr intuitiven Methoden, wie die Perspektiven basierte Lesetechnik, mehr Zeit zum Einarbeiten gebraucht. Außerdem sollten die Perspektiven am besten keine Gemeinsamkeiten aufweisen, wie im Falle von Leser und Autor in unserem Experiment. Auch hätten die Inspektoren besser nur Informatik Studenten sein müssen, da diese sich doch besser zurecht fanden und weniger Schwierigkeiten insgesamt hatten, auch mit dem Vokabular der verschiedenen Wissens-elemente. Besser noch wären die Inspektoren IESE-Mitarbeiter und/oder Fachleute aus den jeweiligen Bereichen.

Auch bei den Wissens-elementenauswahl haben wir Einschränkungen machen müssen. Denn unser Ziel war interne IESE Wissens-elemente und externe Wissens-elemente inspizieren zu lassen. Dazu hätten sich zusätzlich zu den Visek-Dokumenten Defektfälle des Defekt Management Systems Bugzilla unseres Erachtens gut geeignet. Leider hat sich bei der Durchsuchung vieler Defektfälle aus dem Bugzilla

erwiesen, dass die Defektfall/Lösungspaare nicht vollständig waren und damit nicht für die Inspektion geeignet sind. Nichtsdesto Trotz wurden dann insgesamt aus den anderen Bereichen insgesamt acht Wissensdokumente fündig gemacht und inspiziert.

Bei der Inspektion wurden insgesamt 51 Fehler gefunden. 29 Fehler bei Ad-hoc Vorgehen und 22 bei der Perspektiven basierten Methode. Die beim Ad-hoc gefundenen Fehler waren mehr von allgemeiner Natur, während bei der Perspektiven basierten Lesetechnik die Fehler dank der Checklisten besser auf die Wissensdokumente eingingen. Die Tatsache, dass die Anzahl der gefundenen Fehler beim Ad-hoc Vorgehen höher ist als die Anzahl der gefundenen Fehler bei der PBR Methode, hängt nicht zuletzt damit zusammen, dass die Inspektoren aus Zeitgründen bei der PBR-Technik nicht alle Wissens-elemente inspizieren konnten.

Auch wurden hauptsächlich die Fehlerklassen Omission, Miscellaneous, Ambiguous Information und Extraneous gefunden. Das führt zu der Behauptung, dass wenn die Inspektoren Experten auf dem jeweiligen Gebiet gewesen wären, doch die Fehlerklassen Incorrect Fact und Inconsistency vertreten wären.

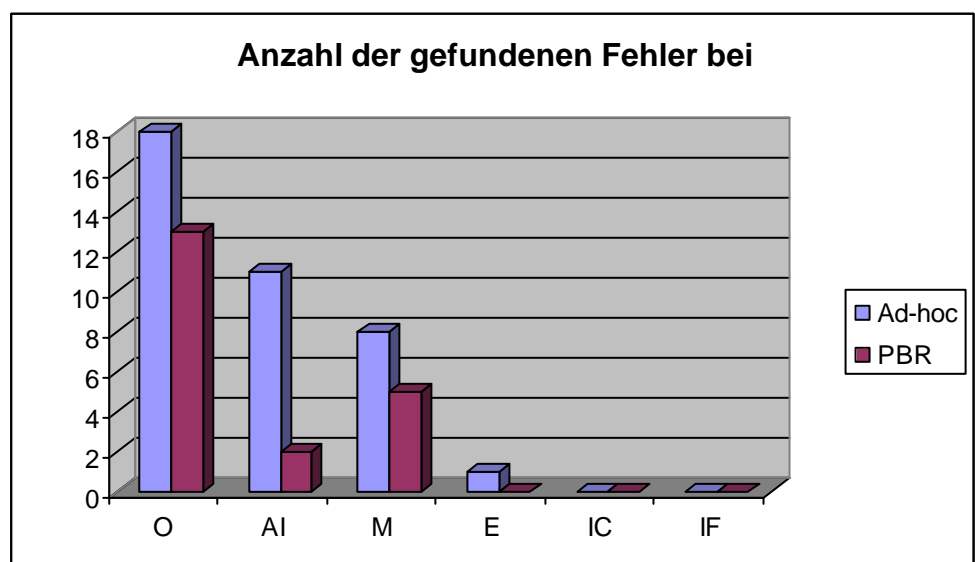


Diagramm 6

Fehlerverteilung bei Ad-hoc und PBR

Die Ergebnissen belegen, dass auf jeden Fall mehrere Leute immer mehr Fehler finden als nur einer, sowohl bei Ad-hoc als auch bei PBR-Methode. Siehe folgende Diagramme:

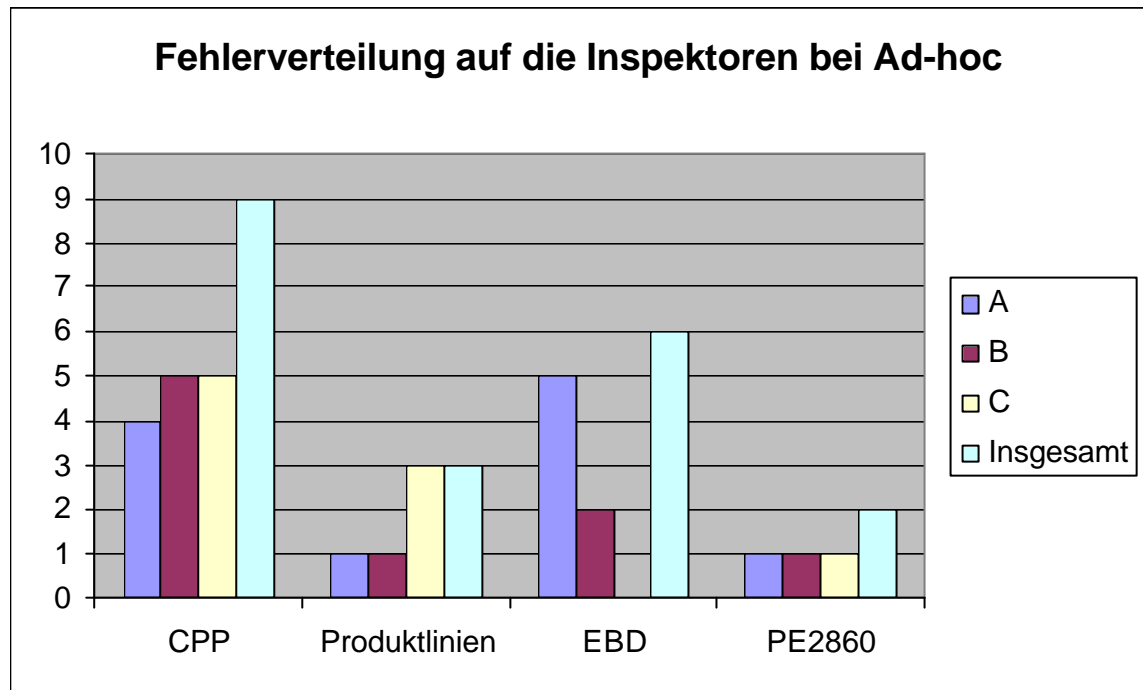


Diagramm 6

Fehlerverteilung auf die einzelnen Inspektoren und insgesamt bei Ad-hoc

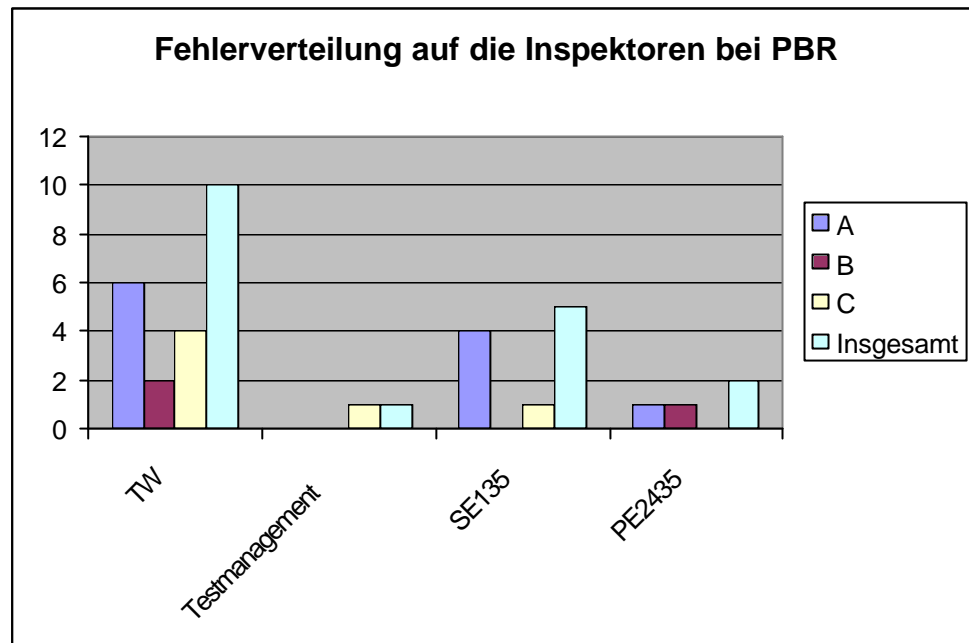


Diagramm 7

Fehlerverteilung auf die einzelnen Inspektoren und insgesamt bei der PBR-Technik

Die Inspektoren haben Fehler gefunden, deren Korrektur die Qualität der Wissensdokumente auf jeden Fall steigern wird. Insgesamt waren die Ergebnisse zufriedenstellend, und sprechen für die Anwendung und Weiterentwicklung der Inspektions-Methoden für Wissensdokumente und Erfahrungen.

5.4 Ausblick

Diese Diplomarbeit liefert einen festen Boden für weitere Arbeiten, diese können vielfältig sein je nachdem wie Schwerpunkte für die einzelnen Arbeiten gelegt werden. So kann sich eine Arbeit beispielsweise mit der Definition von einem allgemeinen Fragenkatalog beschäftigen, in dem alle möglichen Fragen stehen würden, die zur Inspektion von einem Wissensdokument oder einer Erfahrung berücksichtigt werden können. Aus diesem Katalog lassen sich später speziell nach den Bedürfnissen der einzelnen Wissens-elementen maßgeschneiderte Inspektionsfragen schnell auswählen.

Eine andere weiterführende Arbeit könnte sich zum anderen mit den anderen Perspektiven die wir im vierten Kapitel vorgestellt haben beschäftigen. Insbesondere wäre es sinnvoll, die Perspektive Projektmanager bei den Projekterfahrungsdokumenten zu untersuchen.

Auch bietet sich an eine andere Arbeit mit der Definition von allen möglichen Perspektiven durchzuführen, detailliert zu beschreiben und Schnittmengen zwischen den einzelnen Perspektiven so klein wie möglich zu halten.

Danksagung

Ich möchte hiermit Jörg Rech herzlich für seine sinnvolle Betreuung und für seine Kommentare danken, die stets zur Besserung dieser Diplomarbeit beigetragen haben.

Zum Schluss möchten wir uns bedanken bei dem Dozenten der Vorlesung „ERFAHRUNGSMANAGEMENT – ENTWICKLUNG UND EINFÜHRUNG“, Dr. Althoff und seinen Studenten bedanken, die zur Durchführung des Wissensinspektionsexperimentes dieser Diplomarbeit beigetragen haben.

Abbildungsverzeichnis

Abbildung 1	Fehleranzahlen und Fehlerkorrekturkosten	13
Abbildung 2	Klassifikation von Software Prüfverfahren	15
Abbildung3	die drei Phasen einer Inspektion	19
Abbildung4	die sechs Phasen einer Inspektion	21
Abbildung5	die Rollen der Inspektion.....	23
Abbildung 8	Überlappung und Überdeckung innerhalb eines Dokumentes	33
Abbildung 9	Logische Entitäten und deren Dokumentationen	35
Abbildung 10	Lesetechniken	37
Abbildung 11	Beziehung zwischen Daten, Information und Wissen	42
Abbildung 12	Beziehung zwischen Wissen, Intelligenz und Reflexion	44
Abbildung 13	epistemologische Wissensformen	45
Abbildung 14	Das HANSE-Modell	47
Abbildung 15	Die Wissensspirale	49
Abbildung 16	Die Wissensträger	51
Abbildung 17	Die Wissensmodellierungsformen	51
Abbildung 18	Die drei Dimensionen des ganzheitlichen	55
Wissensmanagements		
Abbildung 19	Bausteine des Wissensmanagements.....	57
Tabelle 1	Fehlerklassen	77
Diagramm 2	Fehlerverteilung bei Ad-hoc	91
Tabelle 2	Verteilung der Fehler im Verhältnis zu den Inspektoren.....	92
Diagramm 3	Fehlerverteilung auf die einzelnen Inspektoren und	92
insgesamt		
Tabelle 3	Verteilung der Fehler im Verhältnis zu den Inspektoren.....	94
Diagramm 4	Fehlerverteilung bei PBR.....	95
Diagramm 5	Ad-hoc und PBR im Vergleich.....	96
Tabelle 4	Vor- und Nachteile bei der Ad-hoc Inspektion	97
Tabelle 5	Vor- und Nachteile bei der CLR Inspektion	98
Tabelle 6	Vor- und Nachteile bei der PBR Inspektion.....	98
Diagramm 6	Fehlerverteilung auf die einzelnen Inspektoren und	106
insgesamt bei Ad-hoc		
Diagramm 7	Fehlerverteilung auf die einzelnen Inspektoren und	107
insgesamt bei der PBR-Technik		

Tabellenverzeichnis

Tabelle 1	Fehlerklassen	77
Tabelle 2	Verteilung der Fehler im Verhältnis zu den Inspektoren	92
Tabelle 3	Verteilung der Fehler im Verhältnis zu den Inspektoren	94
Tabelle 4	Vor- und Nachteile bei der Ad-hoc Inspektion	97
Tabelle 5	Vor- und Nachteile bei der CLR Inspektion	98
Tabelle 6	Vor- und Nachteile bei der PBR Inspektion	98

Anhang

Participant-ID: _____**Effort (Minutes):** _____**Dokument :** _____

Defect number	Page	Line # Req. #	Defect class	Short description of defect / Comment
---------------	------	------------------	--------------	---------------------------------------

--	--	--	--	--

--	--	--	--	--

--	--	--	--	--

Checklist for Checklist-Based Reading

How would you rate the following qualities for the given knowledge document (+2: very good / YES, +1 good, ±0 : OK, -1 bad, -2 very bad / NO)

Facet	No.	How to detect	Evaluation		
			+2	+1	0
Clarity	1	Has the knowledge a single, simple, precise and clear goal?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	2	Are the statements in the knowledge document unambiguous ?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Completeness	3	Are the problem, solution, context, purpose, goals, customers/users, constraints, assumptions, and the scope of the experience completely listed in the knowledge document?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	4	Does the knowledge document include all items that are needed for the application or adaptation of the solution (i.e., glossary, links)?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Consistency	5	Are the statements mutually consistent , that is, does any statement in the knowledge document conflict with another statement?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	6	Are the statements consistent with the overview?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Correctness	7	Do the statements describe the problem, solution, ... in a correct manner?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Functionality	8	Are all described steps of the solution/statement necessary and sufficient to meet the objectives of the knowledge product?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	9	Are the inputs to a step of the solution/statement necessary and sufficient to perform the required step?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testability	10	Can the knowledge within be tested, demonstrated or analyzed to show that it satisfies the requirements (solve the problem, reduced work-time, improves product-quality, etc.)?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	11	Is it possible to test the knowledge for itself and not within a greater context (e.g. a real project)?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level of Detail	12	Are the statements free of unnecessary design decisions, i.e., is the solution obscured by too specific / strict preconditions?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	13	Do the statements provide an adequate basis for the application or adaption of the solution?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
--	----	--	-----------------------	-----------------------	-----------------------	-----------------------

7 Subjective / Qualitative Evaluation of knowledge

How would you rate the following qualities for the given knowledge document (+2: very good, +1 good, ±0 : OK, -1 bad, -2 very bad)

Facet	How to detect	Evaluation		
		+2	+1	0
Usefulness of the knowledge	Do you personally think that the knowledge product is useful? Why?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Could you use the knowledge product if you had a related problem? Would you be able to find experts from the document to help you?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Understandability / Acceptance	Did you understand the content of the knowledge? What would you change? What content would you add / remove?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	How specific or general is the content? Could you use it in other environments / in another context?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Significance / ROI	Do you personally think that the knowledge product can be used to save time, train the user, or improve the quality of the product?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	How long will the knowledge encoded be consistent? When should it be reworked / updated?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	Do you personally think that the knowledge product is important (even if not useful)?				
--	---	--	--	--	--

Comments on Knowledge Inspection

How would you rate the following qualities for the given knowledge document (+2: very good, +1 good, ±0 : OK, -1 bad, -2 very bad)

Facet	No.	How to detect	Evaluation
Method	1	What would you change ?	
	2	Are the methods clear and understandable ?	

	3	How much effort (time) did you need for the knowledge documents?	
Knowledge documents	4	Where the documents useful to inspect?	
	5	Other comments?	

Literatur

- [BCR94a] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach,
Experience Factory. In John J. Marciniak, editor,
Encyclopedia of Software Engineering, volume 1
- [Fag76] T. H. Davenport, L. Prusak,
Working Knowledge – How Organisations manage what
they know
School Press, Boston 1998
- [Fag76] Michael E. Fagan,
Design and Code Inspections to reduce Errors in Program
Development.
IBM System Journal
- [Fag86] Michael E. Fagan,
Advances in Software Inspections, IEEE Transactions of
Software Engineering, Vol. SE-12, Nr. 7 Juli 1986
- [FG02] Integriertes Knowledge Management
Konzepte, Methoden, Instrumente, Fallbeispiele
Hrsg. Vieweg & Sohn Verlagsgesellschaft mbH,
Braunschweig/Wiesbaden, 2002
- [Lai00] Oliver Laitenberger,
Cost-effective Detection of Software Defects
through Perspective-based Inspections
Kaiserslautern: Fraunhofer IESE, 2000
- [Lai1a] Oliver Laitenberger,
A survey of Software Inspection Technologies
Kaiserslautern: Fraunhofer IESE
- [Lig02] Peter Liggesmeyer,
Software-Qualität
Berlin: Spektrum, 2002

- [Mül99] Uwe Müller,
Prüf- und Testprozesse in der Softwareentwicklung
Aachen: Shaker Verlag, 1999
- [Non97] Nonaka/Takeuchi,
Organisation des Wissens
New York – Frankfurt/Main 1997
- [Pro97] Gilbert Probst, Kai Romhatdt,
Bausteine des Wissensmanagements – ein
praxisorientierter Ansatz Wiesbaden, 1997
- [Pro98] Gilbert Probst, Steffen Raub, Kai Romhatdt,
Wissen managen: wie Unternehmen Ihre wertvollste
Ressource optimal nutzen . 2.Auflage
Wiesbaden, 1998
- [Rei98] Reinmann-Rothmeier, Mandl,
Wissensmanagement – Eine Delphi-Studie
München, 1998
- [Rom01] Prof. Dr. H. Dieter Rombach,
Software Engineering 1, Vorlesungsfolien
Universität Kaiserslautern, 2001
- [Rus91] G. W. Russel,
Experience with Inspection in Ultralarge-Scale Developpe-
ments, IEE Software, Vol.8. Nr. 1
Januar/Februar 1991
- [Sch00] Peter Schütt,
Wissensmanagement: Mehrwert durch Wissen,
Nutzenpotentiale ermitteln, Den Wissenstransfer organisieren
Falken Verlag, Niedernhausen 2000
- [Sno00] David Snowden,
The ASHEN Model – an enabler of action, in: Knowledge
Management, Band3, Ausgabe 7,
Ark Publications, London 2000

- [Sve97] Karl-Eric Sveiby,
The New Organisational Wealth
San Francisco, 1997
- [Tha96] M. Thaler, M. Utesch,
Effektivität und Effizienz von Softwareinspektionen
Oldenburg 1996
- [Tra96] Prof. Dr. H. Trauboth,
Software Qualitätssicherung
Konstruktive und analytische Maßnahmen
Oldenburg 1996
- [Wah91] Gerhard Wahrig,
Deutsches Wörterbuch,
Bertelsmann Lexikon Verlag GMBH, Gütersloh/München
1991
- [Wil96] Helmut Willke,
Dimensionen des Wissensmanagement -- Zum
Zusammenhang von gesellschaftlicher und organisationaler
Wissensbasierung in Schreyögg/Conrad (Hrsg.):
Wissensmanagement, Berlin -- New York,
1996
- [Wil98] Helmut Willke,
Systematisches Wissensmanagement
Mit Fallstudien von G. Gnewkow
Lucius und Lucius -- Stuttgart, 1998
- [WN86] Winograd/Flores,
Understanding Computers and Congnition
Palo Alto, 1986

