



Universität Karlsruhe (TH)
Fakultät für Informatik
Institut für Telematik
Cooperation & Management



Evaluation von Werkzeugunterstützung bei der Softwareentwicklung

Studienarbeit
von

Tilman Kuhn

Verantwortlicher Betreuer:
Betreuender Mitarbeiter:

Prof. Dr. Sebastian Abeck
Dipl.-Inform. Karsten Krutz

Bearbeitungszeit:
300 Stunden nach Stundenkonto im Zeitraum
November 2004 – April 2005

Dank

Diese Studienarbeit wurde in der Forschungsgruppe Cooperation & Management (C&M) am Institut für Telematik der Fakultät für Informatik an der Universität Karlsruhe verfasst.

Für die Ermöglichung dieser Arbeit bedanke ich mich besonders bei Herrn Professor Sebastian Abeck, da diese Studienarbeit nicht der regulären Art der Studienarbeiten entspricht, die bei C&M verfasst werden, und er sich die Mühe macht in jedem Projektteam aktiv mitzuarbeiten. Ebenso danke ich meinem Betreuer Karsten Krutz, der trotz seines anstrengenden Einsatzes für Informatik I immer die nötige Zeit für Teamtreffen und Privatabsprachen fand und mich mit Informationen, Zeitschriften, Software-CDs, manchmal auch Lions versorgte und einen seiner Laptops als SVN-Server zur Verfügung stellte.

Dank gilt aber auch...

- den weiteren Mitarbeitern von C&M, die mich bei der Studienarbeit unterstützten, insbesondere Christian Emig für seine Beratungen bei der Einrichtung eines Linux-Servers und Nachforschungen über eine EA-Lizenz, Oliver Mehl für seine Vermittlung eines SAP-Softwareentwicklers und allen, die die Pdok des C&MCourseBuilders Korrektur gelesen haben.
- dem restlichen LDoc-Tools-Projektteam (Marco Willsch, Robert Malec, Julia Rohlfing und Florian Maier) für die gute Zusammenarbeit, und dass sie meine Tests immer wieder neuer Software-Werkzeuge ohne Murren mitgemacht haben.
- allen Softwareentwicklern und Mitgliedern der Personalabteilungen, die an meiner Umfrage teilgenommen haben oder für die Teilnahme Sorge trugen.

Mein herzlicher Dank gilt im Besonderen meiner Freundin Claudia Durand für die Geduld, wenn ich öfters mal etwas weniger Zeit hatte und für das Korrekturlesen dieser Arbeit. Ebenso gebührt Dank meiner Familie für das Dach über dem Kopf und die Versorgung mit Liebe, Essen und Elektrizität.

- EINLEITUNG
- Software am Markt
- Der C&M-Softwareentwicklungsprozess
- Unterstützung des Entwicklungsprozesses
- ZUSAMMENFASSUNG UND AUSBLICK

Information 1: EVALUATION VON WERKZEUGUNTERSTÜTZUNG BEI DER SOFTWAREENTWICKLUNG

Inhaltsverzeichnis

1	EINLEITUNG.....	7
1.1	Motivation	7
1.2	Problemstellung	7
1.3	Lösungsansatz	8
1.4	Erzielte Ergebnisse und Gliederung der Arbeit.....	8
2	Software am Markt.....	10
2.1	Software-Kategorien.....	10
2.2	Die Umfrage	11
2.2.1	Ergebnisse tabellarisch.....	12
2.2.2	Ergebnisse textuell.....	13
2.3	Die Recherche	14
2.3.1	Suchbegriffe	14
2.3.2	Web-Portale	14
2.3.3	Ergebnisse	14
2.4	Gesamtergebnis Umfrage und Recherche.....	16
2.5	Vorauswahl	19
2.5.1	Bewertungskriterien	19
2.5.2	Ergebnisse	20
2.5.3	Empfehlung zur weiteren Untersuchung.....	23
3	Der C&M-Softwareentwicklungsprozess.....	24
3.1	Geschäftsprozessmodell.....	24
3.1.1	System modellieren.....	25
3.1.2	System implementieren	28
3.1.3	Projektteam koordinieren	30
4	Unterstützung des Entwicklungsprozesses.....	34
4.1	Generelle Schwierigkeiten	34
4.1.1	Zusammenführen der Ergebnisse.....	34
4.1.2	Zusammenwirken der Werkzeuge	35
4.1.3	Teamarbeit!?	35
4.2	Konkrete Szenarien.....	37
4.2.1	Aktivität „System analysieren“	37

4.2.2	Aktivität „System entwerfen“	39
4.2.3	Aktivität „System implementieren“	42
4.2.4	Aktivität „Aufgaben verwalten“	48
5	ZUSAMMENFASSUNG UND AUSBLICK	50
6	VERZEICHNISSE	52
6.1	Bearbeitete Dokumente	52
6.2	Untersuchte Werkzeuge.....	56
6.3	Abkürzungen und Glossar	56
6.4	Index.....	59
6.5	Informationen.....	59
6.6	Literatur	60

1 EINLEITUNG

1.1 Motivation

Seit ich als Praktikant begonnen hatte, mich im C&M-Projektteam LDoc-Tools einzubringen, vermisste ich einige Software-Werkzeuge, die ich in mehreren Softwareentwicklungsprojekten zuvor schätzen gelernt hatte. Doch warum wurden diese scheinbar nützlichen Produkte, die für mich zu diesem Zeitpunkt aus der Softwareentwicklung nicht mehr wegzudenken waren, bei C&M nicht eingesetzt? Gründe dafür schien es verschiedene zu geben: Man hatte vielleicht einmal schlechte Erfahrung mit einem Werkzeug gemacht (Source Safe), das Produkt machte einfach zu große Eingriffe in die eingeübte Arbeitsweise nötig (Together) oder es war schlicht nicht bekannt, ob und wie das Werkzeug zum eingesetzten Softwareentwicklungsprozess passen könnte. Genau das ist der Punkt, an dem diese Studienarbeit ansetzt.

Stellen Sie sich vor Sie möchten ein Softwaresystem entwickeln. Dabei soll zum einen ein von Ihnen bereits definierter Softwareentwicklungsprozess verwirklicht werden, zum anderen soll die Entwicklung durch geeignete Software-Werkzeuge unterstützt werden, um möglichst schnell und reibungslos an ein gutes Ergebnis zu kommen. Doch wie lassen sich diese beiden Anforderungen an die Softwareentwicklung sinnvoll unter einen Hut bringen? Keinesfalls ist eine Einzelperson dazu in der Lage, wirklich objektiv an diese Fragestellung heranzugehen; ich habe dennoch versucht, diese Untersuchung möglichst aus der Sichtweise der Arbeit im C&M-Projektteam heraus anzugehen.

1.2 Problemstellung

Die Forschungsgruppe Cooperation & Management (C&M) am Institut für Telematik verfolgt Softwareentwicklung nach einem Verfahren, das sich teilweise an [Oe01] orientiert und genauer in [C&M-P-SE] definiert ist. Dieser Entwicklungsprozess soll nun unter Zuhilfenahme geeigneter Software-Werkzeuge so umgesetzt werden, dass die einzelnen Aktivitäten des Geschäftsbereiches der Softwareentwicklung optimal durchgeführt werden können und dabei einen möglichst reibungslosen Gesamt Ablauf ergeben und dass schließlich die von [C&M-PE] 1. Abs. 1 erwähnten „forschungsrelevanten, innovativen Techniken“ nicht nur Gegenstand, sondern teilweise auch Mittel der Softwareentwicklung werden können.

Diese Studienarbeit soll unter anderem eine pragmatische Anleitung geben, wie dieses Ziel erreicht werden kann und welche Software-Werkzeuge dafür wie eingesetzt werden können. Sicher ist in diesem Rahmen keine erschöpfende Analyse sämtlicher am Markt erhältlicher Werkzeuge möglich. Dennoch soll untersucht werden, welche Werkzeuge es gibt und was diese in etwa leisten. Des Weiteren sollen dann einige wenige genauer daraufhin betrachtet werden, wie die verschiedenen Aktivitäten des C&M-Softwareentwicklungsprozesses auf diese abgebildet werden können, um eine einheitliche und dokumentierte Arbeitsweise mit diesen Werkzeugen zu ermöglichen. Dabei spielt auch die Frage danach, ob und wie die erzeugten Ergebnisse und zugehörige erzeugte Dokumentation aus einzelnen Software-Werkzeugen in einem umfassenden Dokument, der C&M-Produktdokumentation integriert werden können, eine wichtige Rolle.

1.3 Lösungsansatz

Um das gestellte Problem anzugehen, muss man zunächst einmal herausfinden, welche Software-Werkzeuge überhaupt für den Einsatz bei C&M in Frage kommen. Dazu benötigt man einen Überblick darüber, welche Werkzeuge der Markt momentan bietet.

Um dies zu erreichen, kann zum einen eine Internetrecherche herangezogen werden, zum anderen ist aber auch eine Analyse der bereits im Fachgebiet eingesetzten Software-Werkzeuge hilfreich. Letzteres könnte durch eine Umfrage bei Unternehmen, die in der Softwareentwicklung aktiv tätig sind, erreicht werden.

Zunächst sollte eine Aufstellung der Software-Werkzeuge gegeben und eine Vorauswahl anhand noch aufzustellender Bewertungskriterien durchgeführt werden. Die Vorauswahl enthält die zur Überprüfung vorgeschlagenen Werkzeuge inklusive der von ihnen bereitgestellten *Features*. Anschließend wurde in Rücksprache mit dem Betreuer und Professor festgelegt, welche Werkzeuge genauer untersucht werden sollten.

Für diese Werkzeuge wurde dann mittels Testeinsatz geprüft, ob und wie sie sich in den gegebenen Softwareentwicklungsprozess integrieren lassen und wie sich die erstellten Teilprodukte und Dokumentationen zur Verwendung in der C&M-Produktdokumentation eignen und zusammenführen lassen. Die Ergebnisse wurden in dieser Ausarbeitung und gegebenenfalls den zu erstellenden Dokumenten für die C&M-T-Map festgehalten.

1.4 Erzielte Ergebnisse und Gliederung der Arbeit

Zunächst wurden in Kapitel 2 die zur bei der Softwareentwicklung eingesetzten Werkzeuge kategorisiert (2.1) und anhand einer Umfrage (2.2) und Internetrecherche (2.3) zunächst überprüft, was es an Werkzeugen der einzelnen Kategorien zur Zeit überhaupt gibt (2.4)¹. Anschließend wurden einige Bewertungskriterien aufgestellt, anhand derer die bereits recht lange Liste an Produkten noch mal betrachtet wurde (2.5). Aus den bewerteten Produkten wurden anschließend die Werkzeuge Powerpoint, Enterprise Architect (EA), Visual Studio .NET 2005 (VS), NDoc und Subversion (SVN) zur weiteren Untersuchung ausgewählt (2.5.3).

Daran anschließend wurde der bei C&M praktizierte Softwareentwicklungsprozess, der ja schon in einigen anderen Dokumenten spezifiziert ist, rekapituliert und in BPMN-Diagrammen veranschaulicht (3).

In der weiteren Durchführung der Studienarbeit wurden die zu untersuchenden Werkzeuge aktiv bei der Entwicklung des LDocEditors eingesetzt. Die dabei gewonnenen Erfahrungen bezüglich der Anwendung der Werkzeuge wurden in den bereits angesprochenen Dokumenten für die C&M-T-Map festgehalten. Diese sind im Einzelnen:

- Modellierungsobjekte für Powerpoint [SA-A-KUHN-10]
- Anwendungstipps für Enterprise Architect [SA-A-KUHN-11]
- Anwendungstipps für Visual Studio .NET 2003 [SA-A-KUHN-13]
- Anwendungstipps für NDoc [SA-A-KUHN-14]

Für Subversion wurde kein explizites Dokument erstellt. Es gibt allerdings bereits ein Buch über den Einsatz von Subversion.²

Weiterhin wurde insgesamt zusammengefasst, wie der Softwareentwicklungsprozess von C&M mit Hilfe der Werkzeuge durchgeführt werden kann. Es wurden grundsätzliche Anforderungen und Schwierigkeiten identifiziert (4.1), die bei der Untersuchung der Prozessunterstützung anhand einzelner Aktivitäten des Prozesses (4.2) nochmals im jeweiligen Kontext beleuchtet werden. Schlussendlich wird noch eine Zusammenfassung und ein Ausblick gegeben (5).

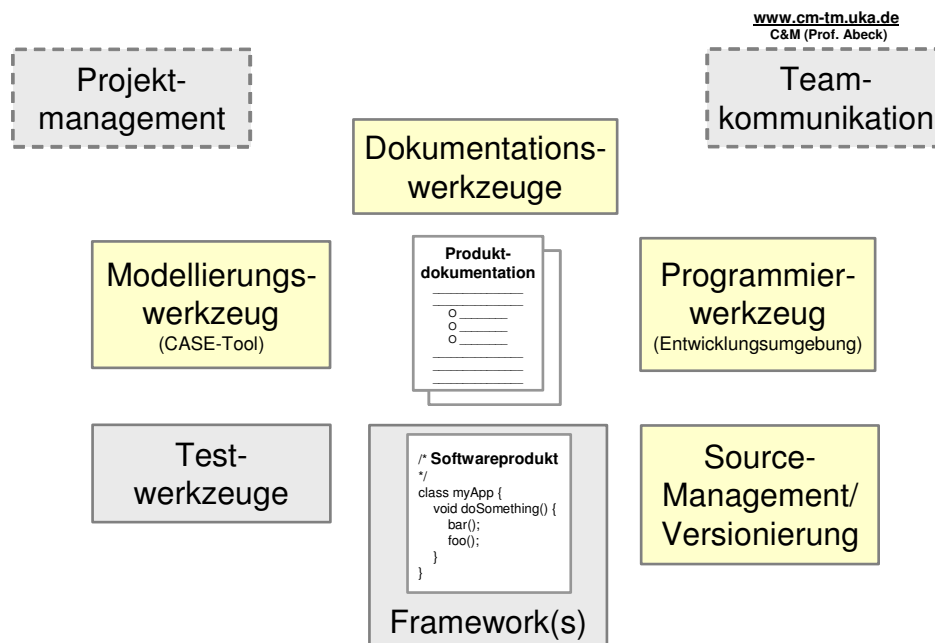
¹ Wobei jedoch bereits nicht alles Gefundene erfasst wurde, sondern bereits eine Selektion des Sinnvollen vorgenommen wurde.

² Und zwar [CF+04]

2 Software am Markt

2.1 Software-Kategorien

Um im Dschungel der Softwarewerkzeuge eine bessere Übersicht zu behalten, sollte man zunächst versuchen, diese in einige sinnvolle Kategorien einzuteilen. Dies wurde im Folgenden versucht. Allerdings ist der Markt hier so breit gestreut, dass diese Liste an Kategorien nicht abschließend sein kann. Des Weiteren lässt sich nicht jedes Softwareprodukt eindeutig einer Kategorie zuweisen, da manche Werkzeuge Bereiche aus mehreren Kategorien abdecken. Die Kategorien Projektmanagement und Teamkommunikation stehen in keinem direkten Bezug zur Softwareentwicklung, sondern sind eher der Teamorganisation zuzurechnen. Sie sind aber der Vollständigkeit halber hier zunächst auch aufgeführt. Die hier grau unterlegten Kategorien sind aus später erläuterten Gründen in dieser Studienarbeit nicht genauer untersucht worden.



Information 2: Software-Kategorien und Dokumente

Projektmanagement

Zu dieser Kategorie zählen Softwarepakete, die die Planung und Durchführung eines Projektes unterstützen, so z.B. auch ERP-Lösungen.

Beispiel: MS-Project

Teamkommunikation

Teamkommunikation kann auf vielfältige Weise unterstützt werden, z.B. durch Onlinekonferenzen, gemeinsame Termin- und Kontaktverwaltung, Wikis und dergleichen. Auch *Trouble-Ticket*-Systeme und *Bug-Tracker* zählen in diese Kategorie. Oftmals werden verschiedene sich ergänzende Produkte in einer so genannten *Groupware*-Lösung kombiniert angeboten.

Beispiele: Bugzilla, eGroupware, ICQ

Modellierung

Programme dieser Kategorie werden zur Erstellung von Modellen der Software verwendet. Dazu zählen u.a. auch die Werkzeuge zur UML-Modellierung.

Beispiele: Together, Enterprise Architect, Poseidon

Programmierung

Unter diesen Titel fallen alle Tools, die zur Erstellung und Wartung von Quellcode „per Hand“ verwendet werden. Also auch die gängigen IDEs.

Beispiele: MS-Visual Studio, Eclipse, NetBeans, Emacs

Test

Spezielle Werkzeuge zum Testen der Software. Dies kann sowohl Software für Akzeptanz- oder *Unit-Tests* sein, als auch solche zur Überprüfung von formal definierten Vor- und Nachbedingungen oder zum Test kompletter Serverapplikationen. Auch *Profiling*-Werkzeuge fallen in diese Kategorie.

Beispiele: JUnit, Cactus, JProfiler

Frameworks

Frameworks sind ein ziemlich weit gefächertes Gebiet. Es gibt sowohl *Frameworks*, die eine komplette Softwareentwicklung abdecken, als auch solche, die nur in einem kleinen Bereich der Komplettssoftware aktiv sind.

Beispiele: J2EE, Struts, .NET, Cocoon, IIS

Source-Management und Versionierung

Die Tools dienen der Verwaltung von Dateisystemen, unterstützen meist die Arbeit als Team an den Dateien und bieten die Möglichkeit, den Entwicklungsprozess der Quellen durch Änderungsprotokollierung nachzuvollziehen und verschiedene Versionen und *Branches* einer Software parallel zu bearbeiten.

Beispiele: CVS, Subversion

Dokumentation

Hier interessiert, was zum Schreiben der Projektdokumentation verwendet wird, d.h. sowohl für solche Dokumente, die sich direkt auf das zu entwickelnde System beziehen, u.a. Pflichtenhefte, UML-Dokumentationen, als auch solche, die begleitend erstellt werden müssen, wie Präsentationen, rechtliche Dokumente, SLAs etc.

Beispiele: JavaDoc, MS-Word, MS-Powerpoint

2.2 Die Umfrage

Um einen, wenn auch aufgrund der Anzahl und Art der erhaltenen Antworten etwas beschränkten, Überblick über die Werkzeuge zu erhalten, die zurzeit in der Geschäftswelt eingesetzt werden, wurde bei einigen Softwareherstellern, die zuvor auf der Bonding-Firmenkontaktmesse diesbezüglich kontaktiert wurden, eine Umfrage durchgeführt. Den dafür verwendeten Fragebogen sowie die Antworten finden Sie in den Anlagen zu dieser Studienarbeit. Das Ergebnis der Umfrage, in dem Antworten von 4 der 10 angeschriebenen Unternehmen zusammengeführt sind, ist in den drei folgenden Tabellen und einer textuellen Zusammenfassung wiedergegeben.

2.2.1 Ergebnisse tabellarisch

Kategorie	CAS Software AG	Hewlett-Packard GmbH, Bereich MSO
Projektmanagement	MS Project CAS genesisWorld	MS Project mit Server
Teamkommunikation	CAS genesisWorld CAS teamWorks ICQ MS Outlook	Serena TeamTrack
Modellierung	MS Visio	Poseidon Rational Rose
Programmierung	Borland Delphi 7	Visual C++ Eclipse IntelliJ Ant
Test	DUnit Pascal Analyzer	JUnit CppUnit TestExpert eigene Entwicklungen
Frameworks	IIS COM+	eigene Entwicklung
Source-Management und Versionierung	MS Visual SourceSafe	ClearCase

Kategorie	IBM Deutschland Entwicklung GmbH (1)	IBM Deutschland Entwicklung GmbH (2)
Projektmanagement	MS Project eigene auf IBM Lotus Domino basierende Tools	MS Project eigene auf IBM Lotus Domino basierende Tools
Teamkommunikation	IBM Lotus Notes	IBM Lotus Sametime IBM WebSphere Portal
Modellierung	Hummingbird Exceed IBM Rational Suite	IBM Rational Suite
Test	eigene Tools	JUnit Mercury LoadRunner eigene auf IBM Lotus Domino basierende Tools
Frameworks	J2EE	J2EE Struts

		JCR 168 JCR 172 J...
Source-Management und Versionierung	eigene Tools	IBM CMVC

Kategorie	SAP AG
Projektmanagement	Eigene Entwicklungen (C-Projekt, C-Arc)
Teamkommunikation	MS Outlook eigene Entwicklungen (internes Online Service System)
Modellierung	Together Visio MS Powerpoint
Programmierung	Eclipse SAP Development Workbench MS VisualStudio
Test	JUnit ABAP Unit-Testing eCat (e Computer Aided Testtool) eigene Entwicklungen
Frameworks	J2EE, WebDynpro NetWeaver
Source-Management und Versionierung	Perforce eigene Tools

2.2.2 Ergebnisse textuell

Klar ist, dass die Ergebnisse keine allzu große Aussagekraft besitzen. Zum einen, weil die Stichprobengröße sehr gering ist und zum anderen weil die zurückgekommenen Umfrageergebnisse die betrachtete Grundgesamtheit der Softwarehersteller nur sehr ungenügend repräsentieren. Bei den tatsächlich teilnehmenden Unternehmen handelt es sich größtenteils um globale Unternehmen, die oftmals in einem Unternehmen integrierte Lösungen herstellen, bei denen die Softwareentwicklung nur als ein Teil von vielen angesehen werden kann. Kleine Software-Unternehmen, die ebenfalls angeschrieben wurden, fanden wohl nicht die Zeit oder das nötige Interesse um an der Umfrage tatsächlich zu partizipieren. Ich versuche dennoch kurz ein paar Erkenntnisse aus den Ergebnissen zusammenzufassen.

Auf der Projektmanagementebene ist MS Project ziemlich weit reichend vertreten, ergänzt durch eigene Tools der Unternehmen. Bei der Kategorie Teamkommunikation sieht die Sache schon ganz anders aus. Hier wird der Microsoft-Lösung mit Outlook und Exchange kein so breites Vertrauen entgegengebracht. Stattdessen kommen meist eigene

Lösungen zum Einsatz. Im Bereich der Modellierung ist von den „billigen“ Office-Werkzeugen bis zu den namhaften Unternehmenslösungen von IBM und Borland alles vertreten. Programmiert wird hauptsächlich mit den gängigen IDEs von Microsoft, Borland und IBM, wobei interessant ist, dass die Sun-eigene Java-IDE NetBeans unter den Teilnehmern nicht zum Einsatz kommt. Was das Quellen-Management anbelangt, verzichtet keines der Unternehmen auf die Vorzüge eines Versionierungssystems, wenn auch die Lizenzpreise für die eingesetzten Systeme beträchtlich sind. Generell ist festzustellen, dass IBM fast durchweg die Werkzeuge aus eigenem Hause verwendet, wie auch zu erwarten war.

2.3 Die Recherche

Zusätzlich zur Umfrage wurde eine mehrtägige Recherche im Internet durchgeführt, um geeignete Software-Werkzeuge zu finden. Dabei kamen die Suchmaschinen Google und Metacrawler mit den unten aufgeführten Suchbegriffen, sowie einige ebenfalls aufgeführte Web-Portale für Entwickler zum Einsatz.

2.3.1 Suchbegriffe

- project management software
- team collaboration software
- bug tracking software
- trouble ticket software
- groupware software
- c# xml documentation comments
- javadoc doclet
- javadoc comments

2.3.2 Web-Portale

- <http://www.project-management-software.org>
- <http://www.thesoftwarenetwork.com>
- <http://www.pmtoday.co.uk>
- <http://www.objectsbydesign.com>
- <http://www.oose.de>
- <http://www.codeassets.com>

2.3.3 Ergebnisse

Die meisten der Softwaresysteme aus der Kategorie Projektmanagement bieten auch Funktionalität der Kategorie Teamkommunikation an und könnten deshalb ohne Probleme unter diesem Punkt nochmals aufgeführt werden. Es sollte dennoch auch nach Lösungen für Teamkommunikation ohne darüber hinaus gehende Projektmanagementunterstützung gesucht werden.

Da es in den meisten Kategorien einen regelrechten Wildwuchs an Software-Werkzeugen gibt, habe ich bereits während der Recherche eine Vorauswahl getroffen und versucht nur einige mir interessant erscheinende Produkte herauszusuchen.

Die Kategorie Frameworks wurde komplett weggelassen. Frameworks spielen zwar in der Softwareentwicklung eine wichtige Rolle, ihre Wahl hängt jedoch stark von den zu erreichenden Ergebnissen der einzelnen Entwicklungsprojekte ab und weniger vom eingesetzten Softwareentwicklungsprozess.

Ebenfalls ausgelassen wurde die Kategorie Test, da das Testen von Software und Dokumenten zum Bereich der Software-Qualitätssicherung gehört und dieses Thema in einer separaten Diplomarbeit von Marco Willsch behandelt wird.

Kategorie	Name	Hersteller
Projektmanagement	ACEProject	Websystems Inc.
	@task	AtTask Inc.
	eProject Enterprise	eProject Inc.
	IT Project Office	Primavera Systems Inc.
	MinuteMan	MinuteMan Systems
	PlanBee Pro	Guy Software
	Project Enterprise	Intellisys Inc.
	Project Management and Help Desk	Autotask Corporation
	TRACK-IT® SUITE	DOVICO Software
	Welcom Open Plan	WST Corporation
	Xcolla™ CD Solution	Axista Inc
	Basecamp	37signals LLC.
Teamkommunikation	bug-track.com	TGMT-systems Inc.
	Bugzilla	mozilla.org
	eGroupWare.org	eGroupware.org
	Lotus Workplace Team Colla.	IBM
	OpenGroupware.org	SKYRIX Software AG
	OPEN-XCHANGE	Open-Xchange.org
	SharePoint Services	Microsoft
	sourceforge.net	Open Source Technology Group
	SourceForge	VA Software Corporation
	TestTrack Pro	Seapine Software Inc.
	Enterprise Architect	Sparx Systems
Modellierung	Magicdraw	No Magic Inc.
	Poseidon	Gentleware
	Rational Software Architect	IBM
	Together	Borland
	Visual Paradigm	Visual Paradigm
	#develop	IC#Code
Programmierung	Delphi	Borland
	Eclipse	eclipse.org
	JBuilder	Borland
	NetBeans	netbeans.org
	Visual Studio .NET	Microsoft
	Aegis	Peter Miller
Source-Management und Versionierung	CVS	CollabNet Inc.
	Subversion	CollabNet Inc.
	Visual SourceSafe	Microsoft
	Doc-o-matic	toolsfactory Software Inc.

Dokumentation	DocFlex/Javadoc	Filigris Works
	DocJet	Tall Tree
	Doxygen	Dimitri van Heesch
	NDoc	Diverse Personen
	Universal Report	Universal Software C. T.

2.4 Gesamtergebnis Umfrage und Recherche

In diesem Abschnitt sind die Ergebnisse aus der Umfrage und Recherche in Tabellen zusammengeführt und durch die URLs und den ungefähren Preis ergänzt.

Die Angaben unter Preis sind nicht direkt vergleichbar, da oftmals sehr unterschiedliche Lizenzierungsmodelle zu Grunde liegen. Daher sind sie mit Vorsicht zu genießen und stellen nur einen groben Anhaltspunkt dar. In den meisten Fällen handelt es sich um den Preis für eine *Named-Lizenz*³.

Die Reihenfolge der Einträge in den Tabellen ist nach den Namen im Alphabet und spiegelt keine Wertung wieder. Eine erste Bewertung gibt es in 2.5.2.

Anmerkungen in den Tabellen:

- (1) Basiert auf Webseiten
- (2) Preis auf Anfrage
- (3) Kostenlos für die Universität Karlsruhe (TH)

Projektmanagement

Name	Hersteller	URL	Preis
ACEProject (1)	Websystems Inc.	http://www.aceproject.com	\$40/ Monat
@task (1)	AtTask Inc.	http://www.attask.com/	(2)
eProject Enterprise (1)	eProject Inc.	http://www.eproject.com/products/enterprise/index.htm	(2)
genesisWorld CRM	CAS Software AG	http://www.cas.de/Produkte/genesisWorld/CRM.asp	(2)
IT Project Office (und andere)	Primavera Systems Inc.	http://www.primavera.com/solutions/po.html	>\$500
Lotus Domino Collaboration Express	IBM	http://www-306.ibm.com/software/de/lotus/produkte/muc.html	€5071/ CPU
MinuteMan	MinuteMan Systems	http://www.minuteman-systems.com/	\$50/ Lizenz
PlanBee Pro	Guy Software	http://www.guysoftware.com/planbee.htm	\$70/ Lizenz
Project Enterprise	Intellisys Inc.	http://www.webintellisys.com/project/enterprise.html	>\$159/ Jahr
Project Management and Help Desk	Autotask Corporation	http://www.autotask.com/	(2)
Project	Microsoft	http://www.microsoft.com/project/	(3)

³ Eine Lizenz die beim Kauf der Software einer eindeutigen Person zugeordnet wird und ausschließlich von dieser verwendet werden darf.

Professional			
TRACK-IT® SUITE (1)	DOVICO Software	http://www.dovico.com/	>\$300
WebSphere Tools	IBM	http://www-306.ibm.com/software/de/websphere/produkte.html	(2)
Welcom Open Plan	WST Corporation	http://www.welcom.com/content.cfm?page=22	(2)
Xcolla™ CD Solution	Axista Inc	http://www.axista.com/	\$100/ Lizenz

Teamkommunikation

Name	Hersteller	URL	Preis
Basecamp (1)	37signals LLC.	http://www.basecamp.com/index.php	\$59/ Monat
bug-track.com	TGMT-systems Inc.	http://www.bug-track.com/	\$985
Bugzilla	mozilla.org	http://www.bugzilla.org/	0
eGroupWare.org	eGroupware.org	http://www.egroupware.org/	0
Lotus Workplace Team Colla.	IBM	http://www.ibm.com/software/lotus/support/workplace-team-collab/index.html	Server +\$90/ Lizenz
OpenGroupware.org	SKYRIX Software AG	http://www.opengroupware.org	0
OPEN-XCHANGE	Open-Xchange.org	http://mirror.open-xchange.org	0
SharePoint Services	Microsoft	http://www.microsoft.com/sharepoint/	(3)
sourceforge.net	Open Source Technology Group	http://sourceforge.net/	0
SourceForge	VA Software Corporation	http://www.vasoftware.com/sourceforge/	(2)
TeamTrack	Serena Software Inc.	http://www.serena.com/Products/teamtrack/home.asp	(2)
teamWorks	CAS Software AG	http://www.cas.de/Produkte/teamWorks/Intranet.asp	ca. €110/ Lizenz
TestTrack Pro	Seapine Software Inc.	http://www.seapine.com/ttpro.html	\$795/ Lizenz

Modellierung

Name	Hersteller	URL	Preis
Enterprise Architect	Sparx Systems	http://www.sparxsystems.com.au/ea.htm	\$225/ Lizenz
Magicdraw	No Magic Inc.	http://www.magicdraw.com/	\$550/ Lizenz
Poseidon	Gentleware	http://www.gentleware.com/	\$69/

			Lizenz
Powerpoint	Microsoft	Über http://office.microsoft.com/	(3)
Rational Software Architect	IBM	http://www-306.ibm.com/software/awdtools/architect/swarchitect/index.html	(2)
Together	Borland	http://www.borland.com/together/	\$5000
Visio	Microsoft	Über http://office.microsoft.com/	(3)
Visual Paradigm	Visual Paradigm	http://www.visual-paradigm.com/vpuml.php	\$700/ Lizenz

Programmierung

Name	Hersteller	URL	Preis
#develop	IC#Code	http://www.icsharpcode.net/OpenSource/SD/	0
Delphi	Borland	http://www.borland.com/delphi/	\$1000/ Lizenz
Eclipse	eclipse.org	http://www.eclipse.org/	0
JBuilder	Borland	http://www.borland.com/jbuilder/	\$500/ Lizenz
NetBeans	netbeans.org	http://www.netbeans.org/	0
Visual SlickEdit	SlickEdit Inc.	http://www.slickedit.com/products/pr_visual_slickedit.php	\$296/ Lizenz
Visual Studio .NET	Microsoft	http://msdn.microsoft.com/vstudio/	(3)

Source-Management und Versionierung

Name	Hersteller	URL	Preis
Aegis	Peter Miller	http://aegis.sourceforge.net	0
CVS	CollabNet Inc.	https://www.cvshome.org/	0
Perforce	Perforce	http://www.perforce.com/	\$750/ Lizenz
Rational ClearCase	IBM	http://www-306.ibm.com/software/awdtools/clearcase/	> \$1700
Subversion	CollabNet Inc.	http://subversion.tigris.org/	0
Visual SourceSafe	Microsoft	http://msdn.microsoft.com/vstudio/previous/ssafe/	\$550 (3)

Dokumentation

Name	Hersteller	URL	Preis
Doc-o-matic	toolsfactory Software Inc.	http://www.doc-o-matic.com/	\$500- \$1000/ Lizenz
DocFlex/ Javadoc	Filigris Works	http://www.filigris.com/products/docflex_javadoc/	€99 (3)
DocJet	Tall Tree	http://www.tall-tree.com/	\$300- \$1000
Doxygen	Dimitri van Heesch	http://www.stack.nl/~dimitri/doxygen/	0
NDoc	Diverse Personen	http://ndoc.sourceforge.net/	0

Powerpoint	Microsoft	Über http://office.microsoft.com/	(3)
Universal Report	Universal Software C. T.	http://www.omegacomputer.com/	\$1000
Word	Microsoft	Über http://office.microsoft.com/	(3)

2.5 Vorauswahl

Bevor mit dem Betreuer konkret festgelegt wurde, welche Software-Werkzeuge genauer unter die Lupe genommen werden sollten, um sie auf die Verwendung im Entwicklungsprozess abzubilden, wurde eine Vorauswahl anhand der unten aufgeführten Bewertungskriterien getroffen. Dies geschah dadurch, dass die auf den Webseiten angegebenen Features entlang der Kriterien bewertet, bzw. gegebenenfalls eine Testinstallation zu Rate gezogen wurde, falls sich die zur Verfügung gestellte Dokumentation als unzulänglich für den notwendigen Überblick herausstellte. Auf alle Fälle wurden keine expliziten Tests durchgeführt, so dass die Ergebnisse als Schätzwerte anzusehen sind. Einige wenige Werkzeuge, deren Einsatz von vorneherein zweifelhaft erschien, wurden dabei nicht in die Bewertung und Vorauswahl miteinbezogen. Darüber hinaus wurden die Kategorien Projektmanagement und Teamkommunikation aus der Vorauswahl ausgeschlossen. Die jeweiligen Gründe sind unten bei der Auflistung der Ergebnisse für die Kategorien vermerkt.

2.5.1 Bewertungskriterien

Im Folgenden eine kurze Erläuterung der Bewertungskriterien, die zur Überprüfung der Werkzeuge verwendet wurden.

Funktionalität

Hauptaugenmerk dieses Kriteriums ist, inwieweit das Software-Werkzeug dazu geeignet wäre, um einen bestimmten Ausschnitt des Softwareentwicklungsprozesses mit ihm durchzuführen. Nebenbei kann auch noch mit einbezogen werden, ob der Einsatz des Werkzeuges einen sonstigen Mehrwert zum Prozess beitragen könnte.

Schnittstellen

Bietet das Werkzeug geeignete Schnittstellen? Hier werden Schnittstellen zu anderen Werkzeugen untersucht, z.B. Schnittstellen zwischen IDE und CASE-Werkzeug oder Schnittstellen zwischen CASE-Werkzeug und Dokumentationswerkzeug. Auch eine mögliche Verbindung zur C&M-Mitgliederverwaltung wäre ein interessanter Aspekt. Der Begriff Schnittstelle ist sehr weit aufzufassen und nicht mit dem Begriff der API-Schnittstellen zu verwechseln. Es könnte sich beispielsweise um Im- und Export in gemeinsamen Dateiformaten oder *Plug-Ins* für das jeweils andere Werkzeug handeln.

Anpassbarkeit

Es kann nicht regelmäßig davon ausgegangen werden, dass ein Software-Werkzeug, auch wenn es sich sinnvoll einsetzen ließe, den C&M-Softwareentwicklungsprozess einhundertprozentig unterstützt. Deshalb muss auch untersucht werden, ob das Werkzeug gegebenenfalls an die speziellen Anforderungen der Softwareentwicklung bei C&M angepasst werden kann. Eine Anpassung meint dabei nicht den Rückgriff auf die werkzeuginternen Konfigurationsmöglichkeiten, sondern eine Erweiterung oder Veränderung des Funktionsumfangs, wie beispielsweise die Entwicklung von *Plug-Ins* oder dergleichen.

Bedienung

Weiterhin von Interesse ist die Bedienung des Werkzeuges. Hat es eine GUI, ist diese übersichtlich und einfach zu benutzen oder lässt es sich gar automatisiert steuern?

Preis

Da der Preis im Normalfall einen wesentlichen Wirtschaftsfaktor darstellt, kann er auch hier nicht ohne Überprüfung bleiben.

2.5.2 Ergebnisse

Die Ergebnisse der meisten Kategorien sind in Tabellen zusammengefasst und enthalten für die Kriterien eine fünfstufige Bewertung (--, -, 0, +, ++).

Projektmanagement

Diese Werkzeuge decken zumeist ein recht breites Spektrum an unternehmerischer Projektorganisation ab, wie sie bei C&M nur in sehr geringen Anteilen betrieben wird. So wird Reporting und Projektkostenkalkulation beispielsweise kaum und Ressourcenmanagement nur sehr eingeschränkt betrieben. Jedenfalls gibt es eine Vielzahl an Möglichkeiten der Softwareunterstützung auf diesem Gebiet, so dass sich die Arbeitsweise von C&M mit Sicherheit auf die eine oder andere Weise unterstützen ließe. Allerdings ist mit solchen Werkzeugen meist ein nicht unerheblicher anfänglicher Einarbeitungsaufwand verbunden, welcher sich nur auszahlt, wenn dadurch mehr erreicht werden kann als danach mit einem Werkzeug dasjenige zu erledigen, wofür zuvor zwei Excel-Tabellen auch ausreichten. Eine Auswahl und Anpassung eines geeigneten Werkzeuges in diesem Bereich könnte in einer eigenständigen Arbeit platz finden, gehört aber nicht zum Kernbereich der Softwareentwicklung und wird daher hier nicht weiter behandelt.

Teamkommunikation

Bei meiner bisherigen Arbeit für C&M vermisste ich am meisten ein www-gestütztes Kommunikationswerkzeug. Vor allem sinnvoll wäre eine von überall zugängliche projekteinheitliche *Todo*-Listenverwaltung. Weitere interessante Ansatzpunkte wären zum einen E-Mail-Listen mit über das WWW zugreifbarer Historisierungsfunktion oder Newsgruppen, so dass alle Teammitglieder auf einfache Weise auf dem Laufenden bleiben können, welche Themen gerade außerhalb der Teamtreffen diskutiert wurden, was bei einfachen persönlichen E-Mails nicht gegeben ist. Aufgrund der Tatsache, dass die Arbeit an dieser Studienarbeit erst gegen Ende des Semesters ihren Höhepunkt erreichte und zur Untersuchung passender Produkte auf die Unterstützung durch die Praktikanten nicht verzichtet werden sollte, war in diesem Bereich bereits zuvor ohne Rücksprache und weitergehende Analyse das Produkt Basecamp von 37signals LLC eingerichtet und zum Testen durch das LDocs-Team freigegeben worden. Die Ergebnisse sind in 4.2.4 zusammengefasst.

Zum anderen ist der Zugriff auf Projektdateien von außerhalb des Uni-Netzes recht mühsam und könnte eventuell verbessert werden: Eine Verbindung mittels VPN und Windows-Freigabe ist selbst mit DSL von der Geschwindigkeit her eigentlich kaum zu gebrauchen und der Zugriff über SSH wird von Windows leider nicht intuitiv unterstützt. Eine Einbindung der C&M-Verzeichnisse als Netzwerkort mittels SSH ist nicht möglich. Vielleicht könnte man in dieser Richtung etwas optimieren. Die bisherige Lösung ist allerdings schon sehr gut im Vergleich zu anderen Instituten. Ansatzpunkte könnten webbasierte Lösungen durch WebDav sein. Für eine einheitliche Dokumentablage empfiehlt sich die Untersuchung von Windows SharePoint Services für den Einsatz bei

C&M in einer getrennten Arbeit. Für die Verwaltung von Quellcodes haben sich die vielfach bewährten Versionierungssysteme durchgesetzt, auf die ich weiter unten noch genauer eingehe. Ansonsten werde ich die Werkzeuge dieser Kategorie nicht weiter verfolgen, da diese auch hier nicht wesentlich auf die eigentliche Softwareentwicklung abstellen.

Modellierung

Name	Hersteller	Funktionalität	Schnittstellen	Anpassbarkeit	Bedienung	Preis
Enterprise Architect	Sparx Systems	+	++	++	0	+
Magicdraw	No Magic Inc.	+	++	++	+	0
Poseidon	Gentleware	0	0	-	-	0
Powerpoint	Microsoft	- -	-	+	+	++
Rational Software Architect	IBM	+	+	++	?	?
Together	Borland	+	++	++	+	- -
Visio	Microsoft	-	0	+	+	++
Visual Paradigm	Visual Paradigm	0	+	0	0	0

Programmierung

Name	Hersteller	Funktionalität	Schnittstellen	Anpassbarkeit	Bedienung	Preis
#develop	IC#Code	-	+	+	+	++
Delphi	Borland	++	+	+	+	--
Eclipse	eclipse.org	++	+	++	++	++
JBuilder	Borland	++	+	+	+	-
NetBeans	netbeans.org	+	0	++	++	++
Visual SlickEdit	SlickEdit Inc.	0	0	0	+	0
Visual Studio .NET	Microsoft	0	0	+	+	++

Source-Management und Versionierung

Name	Hersteller	Funktionalität	Schnittstellen	Anpassbarkeit	Bedienung	Preis
Aegis	Peter Miller	+	-	+	-	++
CVS	CollabNet Inc.	0	+	+	++	++
Perforce	Perforce	+	++	-	+	-
Rational ClearCase	IBM	++	++	-	+	--
Subversion	CollabNet Inc.	+	0	+	++	++
Visual SourceSafe	Microsoft	0	0	0	+	++

Dokumentation

Name	Hersteller	Funktionalität	Schnittstellen	Anpassbarkeit	Bedienung	Preis
Doc-o-matic	toolsfactory Software Inc.	++	++	-	+	--
DocFlex/Javadoc	Filigris Works	++	+	+	+	++
DocJet	Tall Tree	+	+	0	0	--
Doxygen	Dimitri van Heesch	+	0	++	0	++
NDoc	Diverse Personen	+	+	++	+	++
Powerpoint	Microsoft	+	-	+	+	++
Universal Report	Universal Software C. T.	+	+	-	0	--
Word	Microsoft	++	-	+	+	++

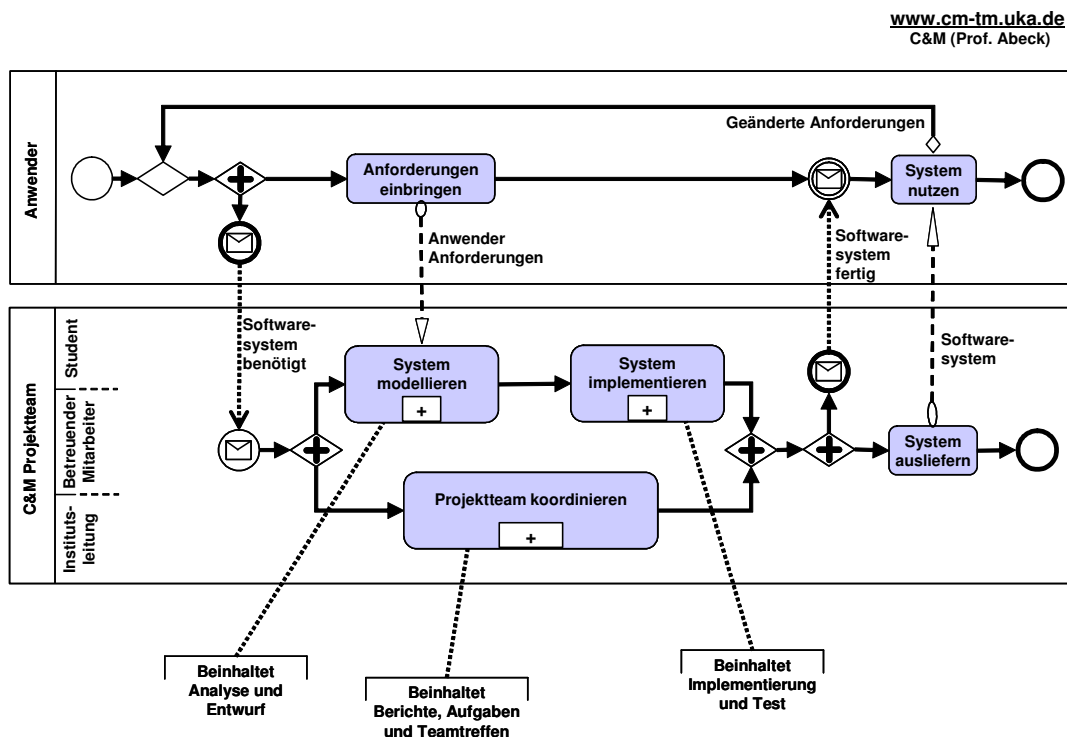
2.5.3 Empfehlung zur weiteren Untersuchung

Die von mir zur weiteren Untersuchung empfohlenen Werkzeuge sind in der obigen Tabelle grau unterlegt. Unter Absprache mit dem Betreuer wurden dann die dunkler unterlegten als tatsächlich zu betrachten festgelegt, da diese zur .NET-Softwareentwicklung im LDocs-Team erprobt werden können. Es sind die Werkzeuge Powerpoint, Enterprise Architect, Visual Studio .NET 2005, NDoc und Subversion.

3 Der C&M-Softwareentwicklungsprozess

Zunächst sollte der von C&M betriebene Softwareentwicklungsprozess genauer unter die Lupe genommen werden. Dazu wurde der Prozess in mehreren Abstraktionsstufen mit BPMN-Diagrammen modelliert. Als Quellen für die Modellierung dienten zum einen die Beschreibung der Softwareentwicklung bei C&M aus [C&M-P-SE] sowie weitere Details zur Systemmodellierung aus [C&M-I-SM], und die bisher erworbenen Erfahrungen aus der Arbeit im C&M-Projektteam LDoc-Tools. Die im Folgenden aufgeführten Diagramme und Erläuterungen stellen also den Ist-Zustand der Softwareentwicklung bei C&M dar.

3.1 Geschäftsprozessmodell



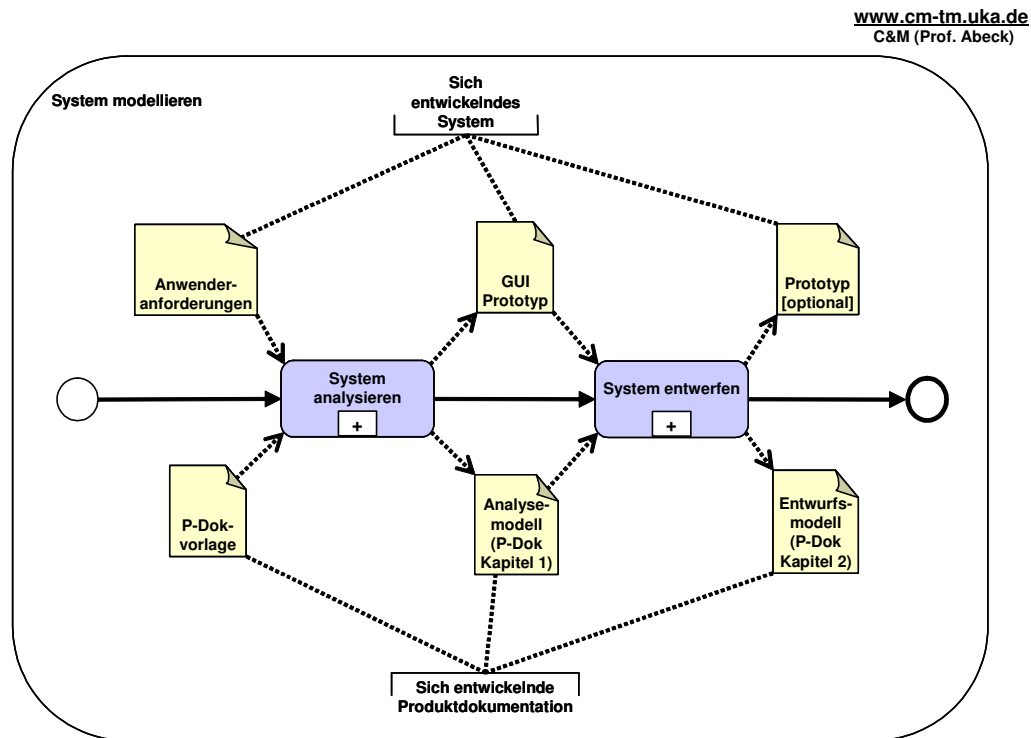
Information 3: Geschäftsprozess Softwareentwicklung bei C&M

[C&M-P-SE] definiert für die Beteiligung an der Softwareentwicklung die drei Rollen Studierende, Mitarbeiter und Leitung, die in dieser Modellierung wieder aufgegriffen sind. Ihre enge Zusammenarbeit soll dadurch zum Ausdruck gebracht werden, dass die entsprechenden *Lanes* im BPMN-Modell nur angedeutet sind und sich sofort wieder verlieren. Jedenfalls ist hier anzumerken, dass zumeist mehrere Personen an der Softwareentwicklung beteiligt sind. Auch die Rolle des Studenten wird oft mehrfach besetzt.

Das Diagramm trägt des Weiteren der von [C&M-P-SE] mehrfach geforderten Auftrennung zwischen der Modellierung des zu erstellenden Softwaresystems und der erst danach folgenden Implementierung Rechnung, indem schon auf dieser sehr abstrakten Ebene zwei getrennte Aktivitäten, System modellieren und System implementieren, modelliert werden. System bezeichnet dabei immer das Softwaresystem. Um eine strukturierte und zielstrebige Softwareentwicklung zu erreichen, ist auch ein gewisses Maß Organisation notwendig. Dazu wurde die Aktivität Projektteam koordinieren ergänzt.

Die Aktivität System nutzen ist nicht dem Geschäftsprozess Softwareentwicklung im eigentlichen Sinne zuzurechnen, sollte aber nicht unerwähnt bleiben und wurde zur Abgrenzung im abgetrennten *Pool* des Softwareanwenders eingefügt. In den folgenden Unterkapiteln wurde auf Beschreibungs- und Diagrammebene in die Unterprozesse eingestiegen, wobei an gegebenen Stellen für genauere Erläuterungen auf die externen Dokumente hingewiesen wird.

3.1.1 System modellieren



Information 4: Unterprozess System modellieren BPMN-Diagramm

Der Geschäftsprozess System modellieren setzt sich aus den beiden Aktivitäten System analysieren und System entwerfen zusammen, in denen das Softwaresystem und die zugehörige Produktdokumentation fortentwickelt und verfeinert werden.

System analysieren

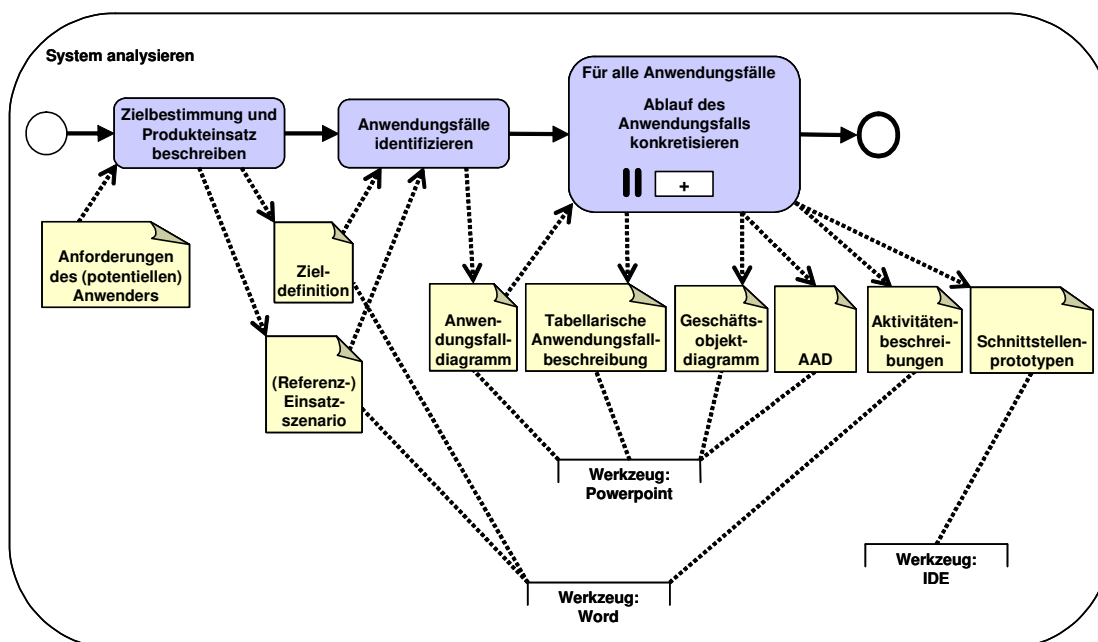
Die Analysephase ist in [C&M-I-SM] Kapitel 2 genau beschrieben und wurde hier nochmals in einer tabellarischen Übersicht und einem BPMN-Diagramm, das den Inhalt der ursprünglichen Information 10 wiedergibt, zusammengefasst.

www.cm-tm.uka.de
C&M (Prof. Abeck)

Kurzbeschreibung:	„Erfassung der Anforderungen des Kunden an das System. Da der Kunde diese Anforderungen definiert, gibt das aus der Analyse erzielte Ergebnis die Sicht des Kunden auf das Produkt wieder.“
Vorbedingungen:	Die Produktdokumentationsvorlage steht zur Verfügung
Eingehende Infos:	P-Dok-Vorlage, Anforderungen des (potenziellen) Anwenders
Ergebnisse:	(Referenz-) Einsatzszenario, Zieldefinition, Anwendungsfalldiagramm, tabellarische Anwendungsfallbeschreibung, AADs, Aktivitätenbeschreibungen, Geschäftsobjektdiagramm, Schnittstellenprototypen, (optional) 1. Prototyp
Nachbedingungen:	Die Anforderungen des Anwenders sind verstanden und erfasst worden. Analyseergebnisse sind ausreichend in die Produktdokumentation übertragen.
Ablauf:	1 Zielbestimmung und Produkteinsatz beschreiben 2 Anwendungsfälle identifizieren 3 Abläufe der Anwendungsfälle konkretisieren

Information 5: Unterprozess System analysieren Beschreibung

www.cm-tm.uka.de
C&M (Prof. Abeck)



Information 6: Unterprozess System analysieren BPMN-Diagramm

In den *Annotations* ist zu sehen, welche *Data Objects* mit welchem Werkzeug hergestellt werden. Alle *Data Objects* zusammengekommen ergeben das Analysemodell aus Information 4 und fließen also in die Produktdokumentation im Kapitel 1 ein.

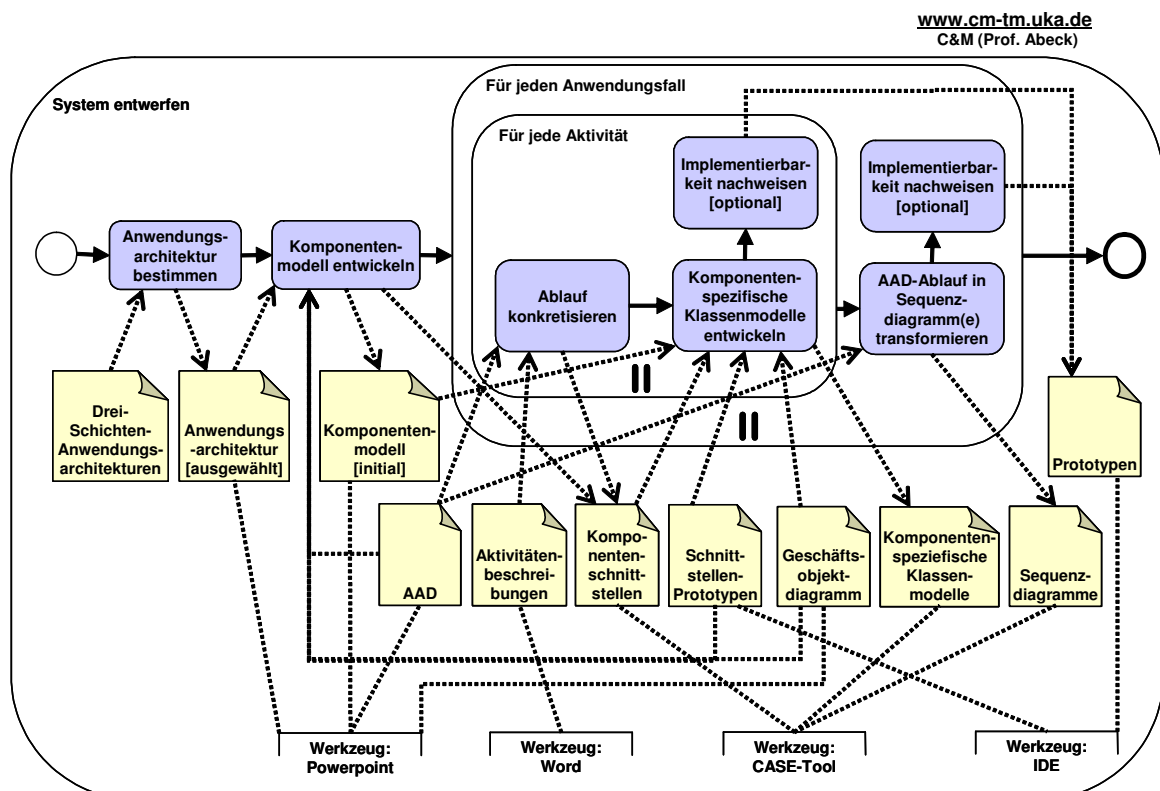
System entwerfen

Auch die Entwurfsphase ist in [C&M-I-SM] beschrieben (Kapitel 2). Hier die Zusammenfassung und ein paar Ergänzungen:

www.cm-tm.uka.de
C&M (Prof. Abeck)

Kurzbeschreibung:	„Das Analysemodell wird umgeformt, so dass das Ergebnis den technischen Anforderungen gerecht wird. Es entsteht das Entwurfsmodell und die Softwarearchitektur als Teil davon.“
Vorbedingungen:	Die Ergebnisdokumente einer vorangegangenen Analysephase liegen vor.
Eingehende Infos:	Drei-Schichten-Anwendungsarchitekturen, AADs, Aktivitätenbeschreibungen, Schnittstellenprototypen, Geschäftsobjektdiagramm
Ergebnisse:	Anwendungsarchitektur, Komponentenmodell, Komponentenschnittstellen, komponentenspezifische Klassenmodelle, Anwendungsfall-, Sequenzdiagramme, Prototypen
Nachbedingungen:	„Nach dem Entwurf sollte man verstanden haben, wie die Lösung aussieht.“ Entwurfsergebnisse sind ausreichend in die Produktdokumentation übertragen.
Ablauf:	1 Anwendungsarchitektur bestimmen 2 Komponentenmodell entwickeln 3 Ablauf der Aktivitäten konkretisieren 4 Komponentenspezifische Klassenmodelle entwickeln 5 AAD-Ablauf in Sequenzdiagramm(e) transformieren 6 Implementierbarkeit von Teilmodellen des Entwurfs nachweisen [optional]

Information 7: Unterprozess System entwerfen Beschreibung



Information 8: Unterprozess System entwerfen BPMN-Diagramm

Das Originaldiagramm aus [C&M-I-SM] Information 31 enthält um die Übersichtlichkeit zu bewahren keine Geschäftsobjekte. Information 8 zeigt denselben Ablauf in BPM-Notation, jetzt jedoch ergänzt um die *Data Objects*, damit nachvollzogen werden kann, in welchen Aktivitäten die Objekte aus der Analyse verwendet werden, welche Artefakte neu entstehen und mit welchen Werkzeugen am Entwurfsmodell gearbeitet wird. Das Entwurfsmodell umfasst die in diesem Unterprozess neu entstehenden *Data Objects*.

3.1.2 System implementieren

Nach dem Entwurf sind die Modelle des Softwaresystems so weit verfeinert, dass mit der verbleibenden Implementierung und dem Testen begonnen werden kann. Dies ist in [C&M-I-SM] Kapitel 4 nur sehr kurz angerissen; so ist für die Implementierung, abgesehen von drei Forderungen aus [C&M-I-SM] Information 48 bisher keine weitere Standardisierung vorgesehen.

Zur vollständigen Abdeckung des gesamten Prozesses der Softwareentwicklung wurden aber hier die letzten Schritte etwas ausführlicher beschrieben und ebenfalls mittels BPMN modelliert. Die genannten Forderungen

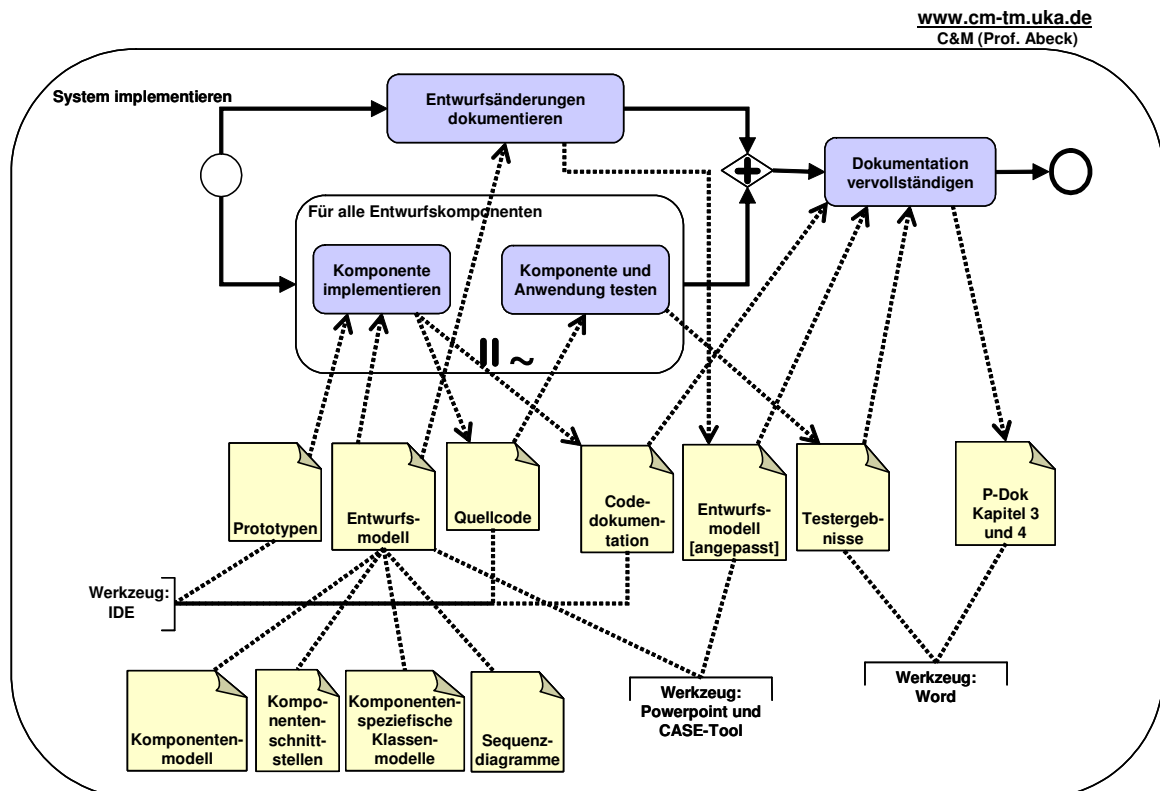
- Erfahrungen der erstellten Prototypen mit einbeziehen
- Änderungen am Modell im Entwurf aktualisieren
- Implementierung gut dokumentieren!

wurden dabei in der tabellarischen Beschreibung in den Bereichen Eingehende Infos, Nachbedingungen und Ergebnisse mit eingeflochten.

www.cm-tm.uka.de
C&M (Prof. Abeck)

Kurzbeschreibung:	Überführung des Entwurfsmodells in ein maschinenausführbares Programm mit zugesicherten Qualitätsmerkmalen und Eigenschaften. Dabei auch Anpassungen im Entwurfsmodell wo diese nötig sind.
Vorbedingungen:	Die Ergebnisdokumente einer vorangegangenen Entwurfsphase liegen vor.
Eingehende Infos:	Entwurfsmodell (Komponentenmodell, Komponentenschnittstellen, Komponenten-spezifische Klassenmodelle, Sequenzdiagramme, inklusive zugehöriger Dokumentation), Prototypen
Ergebnisse:	Angepasstes Entwurfsmodell, Quellcode des Produktes, Codedokumentation, Dokumentierte Testergebnisse, PDok Kapitel Implementierung und Einsatz
Nachbedingungen:	Das Softwareprodukt ist lauffähig und ausreichend getestet. Die Dokumentation für Implementierung, Tests und Einsatz ist komplett. Das Entwurfsmodell und die zugehörige Dokumentation ist konsistent mit der tatsächlichen Implementierung.
Ablauf:	1 Entwurfskomponente implementieren 2 Entwurfsänderungen dokumentieren 3 Komponenten und Anwendung testen 4 Dokumentation vervollständigen

Information 9: Unterprozess System implementieren Beschreibung

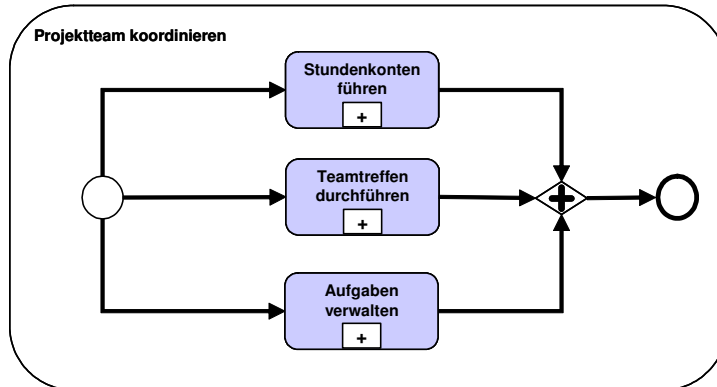


Information 10: Unterprozess System implementieren BPMN-Diagramm

Der in Information 10 visualisierte Prozess sieht zwar eine iterative oder parallele Implementierung der einzelnen Komponenten vor, ist aber in dieser Form nur als Orientierungshilfe für die zu erledigenden Aktivitäten zu sehen. Was die Tests angeht, ist dem Verantwortlichen durch den *Ad-Hoc Marker* ohnehin weitgehende Freiheit eingeräumt. Die hier nicht geänderte Standardreihenfolge für *Multiple Instance Loops* ist zwar die sequenzielle Ausführung, es ist mitunter jedoch durchaus sinnvoll, mehrere Komponenten parallel zu implementieren, vor allem wenn dies durch verschiedene Teammitglieder geschehen kann. Als Hauptaugenmerk sind auch in diesem Diagramm wieder die in den einzelnen Schritten verarbeiteten und erzeugten *Data Objects* von Interesse. In den Prozess eingehende Artefakte sind wie zu erwarten die Ergebnisse aus dem Entwurf, die hier andeutungsweise unter Entwurfsmodell zusammengefasst werden können. Ergebnisse des Prozesses sind zum einen das Softwaresystem selbst, also der Quellcode, aber auch die zugehörige Dokumentation in Form der Codekommentare und der Pdok. Auch die Ergebnisse der Tests, vor allem der kritischen Fälle sind in die Dokumentation mit aufzunehmen.

3.1.3 Projektteam koordinieren

www.cm-tm.uka.de
C&M (Prof. Abeck)



Information 11: Unterprozess Projektteam koordinieren BPMN-Diagramm

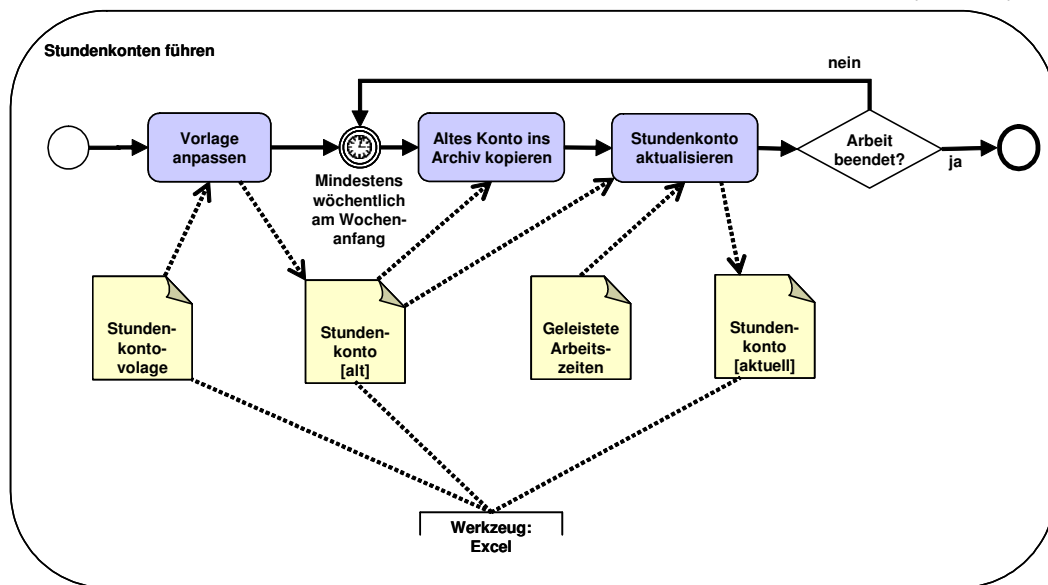
Die Koordination der Projektteams zerfällt in drei im Wesentlichen parallel ablaufende Tätigkeiten: Das Führen von Stundenkonten, die periodisch stattfindenden Teamtreffen und die Verwaltung der teaminternen Aufgabenverteilung.

Stundenkonten führen

www.cm-tm.uka.de
C&M (Prof. Abeck)

Kurzbeschreibung:	Es werden mindestens wöchentlich aktualisierte Stundenkonten geführt, die in den Teamtreffen eingesehen werden können.
Vorbedingungen:	Die entsprechenden Vorlagen stehen zur Verfügung.
Eingehende Infos:	Stundenkontovorlage, geleistete Arbeitszeiten
Ergebnisse:	Kontinuierlich aktuelle Stundenkonten
Nachbedingungen:	Arbeitszeit bei C&M erreicht und dokumentiert
Ablauf:	1 Vorlage anpassen 2 Altes Konto ins Archiv kopieren 3 Stundenkonto aktualisieren

Information 12: Unterprozess Stundenkonten führen Beschreibung



Information 13: Unterprozess Stundenkonten führen BPMN-Diagramm

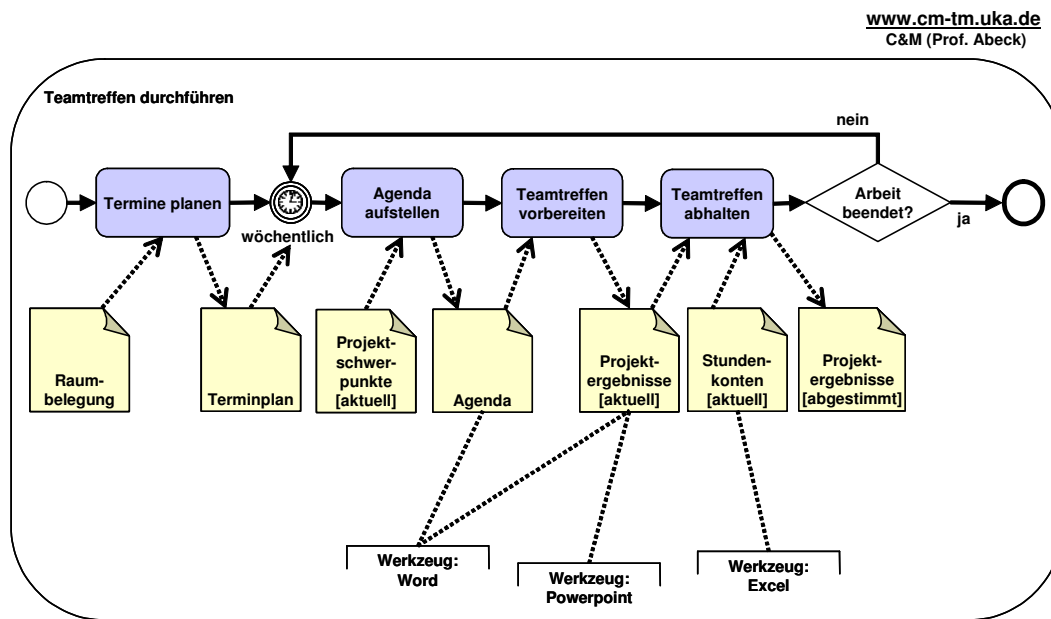
Die Stundenkonten sollen laut [C&M-P-AW] Kapitel 4.1 zur „Gewährleistung einer zielgerichteten und konsequenten Durchführung“ der studentischen Mitarbeit in den Projektteams beitragen. Dazu müssen sie zumindest am Anfang jeder Woche auf den aktuellen Stand gebracht werden, damit während der Teamtreffen auf die Arbeitszeiteinformationen Bezug genommen werden kann. (Dazu Information 15). Um eine einheitliche und reibungslose Arbeit mit den Stundenkonten zu ermöglichen, wird eine mit Makros ergänzte Excel-Vorlage bereitgestellt.

Teamtreffen durchführen

Ziele und Durchführung der (während der Vorlesungszeit) wöchentlich stattfindenden Teamtreffen wird in [C&M-P-AW] Kapitel 3.1 beschrieben und wurde hier nochmals in einer Tabelle und einem BPMN-Diagramm zusammengefasst.

Kurzbeschreibung:	Wöchentlich findet ein Treffen des Projektteams statt „zur Abstimmung der gemeinsamen Projektarbeit und zur inhaltlichen Diskussion auf der Basis der erarbeiteten Ergebnisse“.
Vorbedingungen:	Die aktuell erarbeiteten Ergebnisse liegen vor den jeweiligen Treffen in einer anschaulichen Aufbereitung (beispielsweise in der Produktdokumentation) vor und die Stundenkonten sind auf dem aktuellen Stand.
Eingehende Infos:	Raumbelegung, aktuelle Schwerpunkte, aktuelle Ergebnisse, aktuelle Stundenkonten
Ergebnisse:	Terminplan, Agenda jeweils pro Woche, abgestimmte Projektergebnisse
Nachbedingungen:	Abgestimmtes teamorientiertes Arbeiten ist möglich, die Junior-Students sind ausreichend miteinbezogen.
Ablauf:	1 Termine planen 2 Agenda aufstellen 3 Teamtreffen vorbereiten 4 Teamtreffen abhalten

Information 14: Unterprozess Teamtreffen durchführen Beschreibung



Information 15: Unterprozess Teamtreffen durchführen BPMN-Diagramm

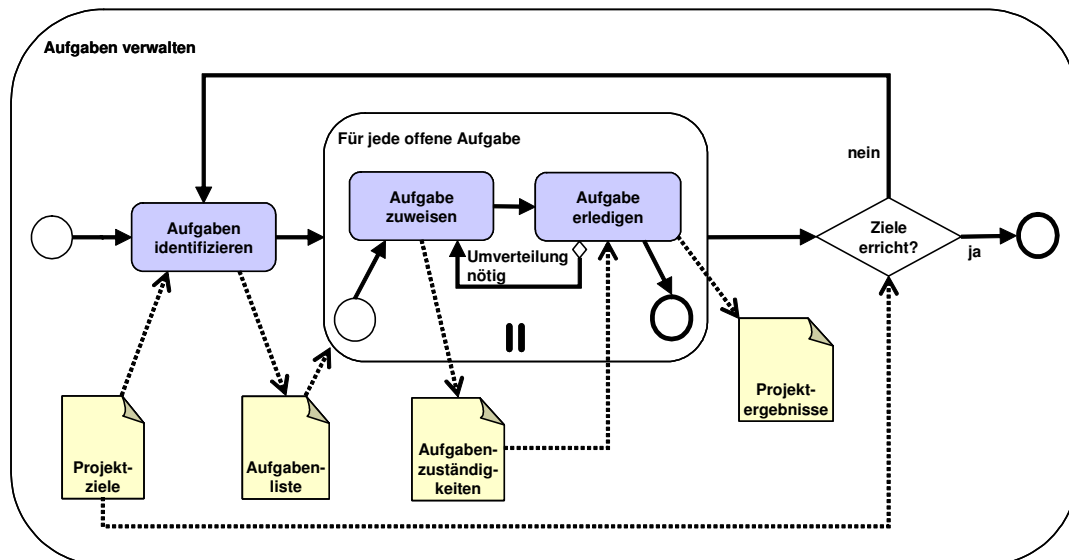
Nach einem am Semesterbeginn festgelegten Terminplan werden wöchentlich Teamtreffen abgehalten. Auf die Agenda kommen Themen, die sich auf die aktuellen Projektschwerpunkte der Teilnehmer beziehen. Die Teilnehmer sind gebeten, sich auf die Treffen vorzubereiten, so dass die aktuellen Projektergebnisse und eventuell organisatorischen Dinge zielgerichtet besprochen werden können. Zusätzlich sollen die Studierenden ihre aktuellen Stundenkonten zur Verfügung halten, um ihre letzten Tätigkeiten nachweisen und erläutern zu können.

Aufgaben verwalten

Für eine effektive Teamarbeit ist es erforderlich, die aktuell anstehenden und die aufkommenden Aufgaben zu verwalten, aber eingeschränkt auch die bereits geleisteten zu archivieren. Die Verwaltung von Aufgaben innerhalb der einzelnen Projektteams ist bei C&M bisher durch kein Profildokument einheitlich geregelt, sondern wird von den unterschiedlichen Teams in Eigenregie durchgeführt. Die wenigen Grundprinzipien, die hinter einer vernünftigen Aufgabenplanung stecken, sind jedoch immer dieselben und in der Beschreibung und dem Diagramm unten wiedergegeben.

Kurzbeschreibung:	Zum Erfolg eines Projektes sind viele Einzelschritte von Nöten. Diese Einzelaufgaben müssen erkannt, verteilt und abgearbeitet werden.
Vorbedingungen:	Zu erreichende Projektziele sind klar genug definiert und abgegrenzt.
Eingehende Infos:	Projektziele
Ergebnisse:	Aufgabenliste, Aufgabenzuständigkeiten, Projektergebnisse
Nachbedingungen:	Die Projektziele werden erreicht, erforderliche Zwischenschritte werden nicht vergessen.
Ablauf:	1 Aufgaben identifizieren 2 Aufgaben zuweisen 3 Aufgaben erledigen

Information 16: Unterprozess Aufgaben verwalten Beschreibung



Information 17: Unterprozess Aufgaben verwalten BPMN-Diagramm

Der Grund, eine Aufgabenverwaltung anzustrengen, ist in der Regel immer der Wunsch, die Ziele des Projektteams zu erreichen. Um die Möglichkeit zu schaffen Einzelaufgaben zu identifizieren, die das Team gemessen an den Zielen weit voran bringen, müssen die Ziele klar genug definiert und abgegrenzt sein. Der Ablauf für jede erkannte Aufgabe selbst ist im Grunde recht einfach: Es muss ein Verantwortlicher gefunden werden, der diese Aufgabe erledigt. Sollte die Erledigung für ihn nicht möglich sein, muss die Aufgabe weiterdelegiert werden, wodurch die Zuständigkeit einem neuen Verantwortlichen übertragen wird. Der ganze Prozess endet, wenn die Ziele des Projektes erreicht sind und folglich keine neuen Aufgaben mehr zu identifizieren sind.

4 Unterstützung des Entwicklungsprozesses

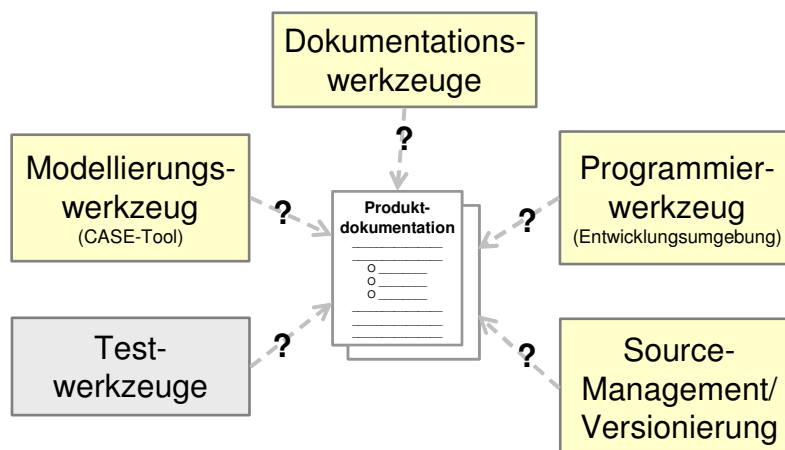
In diesem Kapitel werden nun die möglichen Ansatzpunkte und Ergebnisse aufgezeigt, wie man Ausschnitte des im vorigen Kapitel beschriebenen Softwareentwicklungsprozesses von C&M mit den zur Untersuchung bestimmten Werkzeugen durchführen kann. Da wie zu erwarten nicht alles immer harmonisch laufen kann, werden positive Seiten ebenso wie negative Seiten aufgezeigt, doch dabei wurde nicht sämtliche Dokumentation, die in dieser Studienarbeit entstand, auch in dieser Datei abgelegt. Einiges ist auch in die den einzelnen Werkzeugen zuzuordnenden Dokumente für die C&M-T-Map eingeflossen, welche als Anlagen zur Verfügung stehen. Dies wird an den entsprechenden Stellen nochmals genauer erwähnt.

Doch bevor es ins Detail geht, wurden einige generelle Probleme, die bei den Untersuchungen und der Arbeit mit den Werkzeugen im Team auftraten, identifiziert und zusammengefasst.

4.1 Generelle Schwierigkeiten

4.1.1 Zusammenführen der Ergebnisse

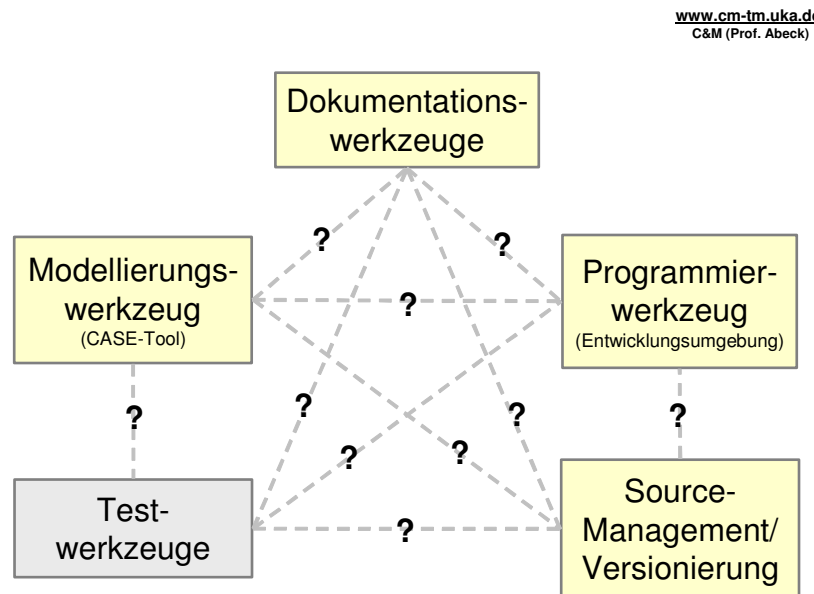
www.cm-tm.uka.de
C&M (Prof. Abeck)



Information 18: Zusammenführen der Ergebnisse

Zentrales Dokument der Softwareentwicklung bei C&M ist laut [C&M-P-SE] und weiter spezifiziert in [C&M-I-SM] die auf Microsoft Word, und Microsoft Powerpoint und den Regeln aus [C&M-P-KE] basierende C&M-Produktdokumentation. In ihr werden die Ergebnisse der einzelnen Phasen der Softwareentwicklung dokumentiert und zusammengefasst. Werden die Ergebnisse jedoch mit einzelnen Werkzeugen der Kategorien aus Kapitel 2.1 erbracht, so stellen sich die folgenden Fragen: Wie bekommt man die Ergebnisse in die Produktdokumentation? Muss alles per Hand abgetippt/abgezeichnet werden oder können zumindest Dokumentationsteile aus meinen Werkzeugen herauskopiert werden? Kann Dokumentation generiert werden? Wie kann die Dokumentation bei Änderungen außerhalb aktuell gehalten werden? Diese Fragen sind für die einzelnen Werkzeuge zu klären.

4.1.2 Zusammenwirken der Werkzeuge



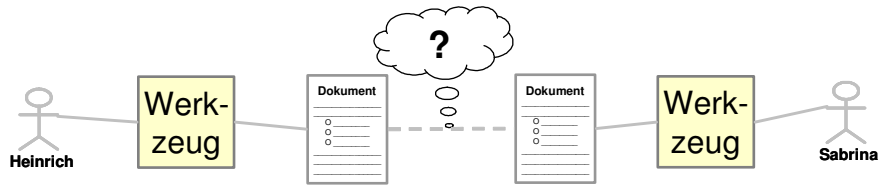
Information 19: Zusammenwirken der Werkzeuge

Bei der Softwareentwicklung wird mit einer heterogenen Umgebung von Werkzeugen gearbeitet. Dabei kommen in verschiedenen Phasen verschiedene Werkzeuge zum Einsatz. Die Arbeit einer Phase baut aber auf die Ergebnisse der vorangegangenen Phase auf, wie aus Information 4 und Information 10 ersichtlich. Auch hier stellt sich wieder die Frage, wie die Ergebnisse des einen Werkzeugs als Quellen des nächsten Werkzeugs verwendet werden können. Welche Verbindungen können hergestellt werden? Gibt es gemeinsam genutzte Formate, oder können die Werkzeuge gar miteinander interagieren? Welche Kommunikationsrichtungen werden unterstützt? Welche Verbindungen sind überhaupt sinnvoll? Ein Beispiel für ein konkretes Szenario wäre die Interaktion zwischen Modellierungswerkzeug und der Entwicklungsumgebung. Die meisten Modellierungswerkzeuge können Quelltext, zumindest eine Ausgangsversion, generieren. (*source code generation*) Bessere Werkzeuge können zusätzlich am Quelltext vorgenommene Änderungen wieder in das Entwurfsmodell übernehmen und damit Modell und Programm konsistent halten (*round trip engineering*).

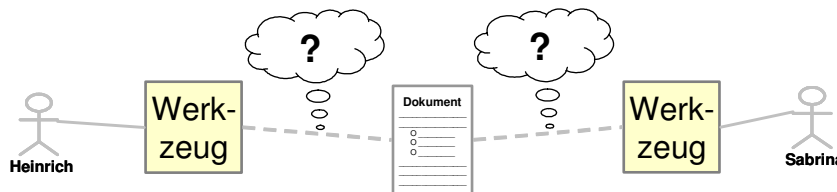
4.1.3 Teamarbeit!?

Aus [C&M-P-AW] geht hervor, dass ein wichtiger Aspekt der Arbeit bei C&M das Wirken im Team ist. Dies gilt insbesondere auch für die Softwareentwicklung. Hier arbeiten zumeist mehrere Studierende und Betreuer an einem gemeinsamen Projekt. Zwar nicht zwangsläufig immer gerade an ein und demselben Dokument, doch auch dieser Fall kommt mitunter vor, da ja ein gemeinsames Ergebnis angestrebt wird. Müssen nun verschiedene Personen an denselben Informationen arbeiten, sind prinzipiell zwei Szenarien denkbar:

1. Arbeiten auf getrennten Dokumenten



2. Arbeiten auf einem gemeinsamen Dokument



Information 20: Teamarbeit!? Zwei Szenarien

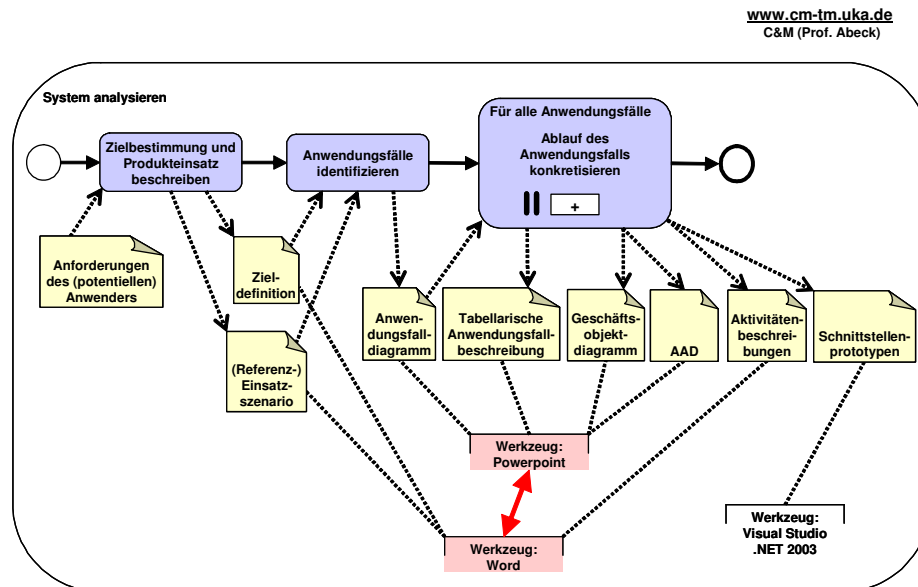
Im 1. Fall arbeiten Heinrich und Sabrina jeweils an einer eigenen Version eines Dokuments; jeder nimmt Änderungen an seiner privaten Version vor. In dem Moment, in dem ein gemeinsames Ergebnis erzielt werden soll, müssen die beiden Dokumente verschmolzen werden, wobei keine Änderungen verloren gehen sollten. Diese Verschmelzung bedeutet unter Umständen einiges an Arbeitsaufwand, vor allem dann, wenn sie zur Synchronisation der Dokumente häufig vorgenommen wird. Das Fragezeichen deutet im Kontext dieser Arbeit an, ob es nicht Möglichkeiten gibt den Prozess der Fusionierung der Dokumente durch die eingesetzten Werkzeug zu unterstützen.

Der 2. Fall sieht etwas anders aus. Hier bekommt nicht jeder sein eigenes Dokument, sondern Heinrich und Sabrina arbeiten auf ein und demselben Datensatz. Von Hand ist dies in der Regel nur dann möglich, wenn die beiden nicht gleichzeitig am Dokument arbeiten, sondern zu jedem Zeitpunkt ausschließlich eine Person das Recht hat, die nötigen Manipulationen vorzunehmen. Auch hier ist interessant, wie der Zugriff auf ein gemeinsames Dokument geregelt werden kann und welche Werkzeugunterstützung es dafür gibt. Möglicherweise gibt es sogar Fälle, in denen echt gleichzeitig gearbeitet werden kann.

Unabhängig davon, welches Szenario für welches konkrete Dokument zum Einsatz kommt, kann man davon ausgehen, dass eine sinnvolle Werkzeugunterstützung hier einen wesentlichen Mehrwert zur Teamarbeit durch Zeitersparnis leistet.

4.2 Konkrete Szenarien

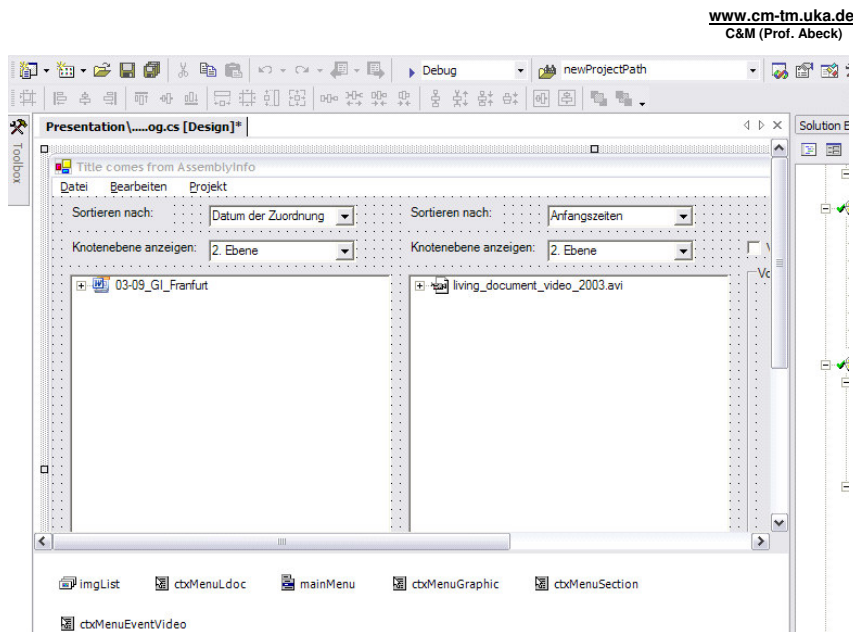
4.2.1 Aktivität „System analysieren“



Information 21: Aktivität System analysieren

Bei der Analyse eines zu entwickelnden Softwaresystems werden, unter Interaktion mit dem Kunden und Anwendern der Software, die Anforderungen und Anwendungsfälle der Software erfasst. Das Modellierungswerkzeug Enterprise Architect unterstützt auch die hierbei eingesetzten Diagrammtypen. Dennoch ist es aus verschiedenen Gründen auf diesem hohen Abstraktionsniveau empfehlenswert, auf ein spezielles Modellierungswerkzeug zurückzugreifen. Zum einen steht man ja in Kundeninteraktion und möchte den Kunden mit einem komplexen Werkzeug nicht überfordern, vor allem wenn er sich eventuell schon an die ihm bisher unbekannte Notation der UML gewöhnen muss. Daher ist die Verwendung allseits bekannter Office-Werkzeuge ratsam, zumal dann auch in Sitzungen auf einfache Weise an den Diagrammen gearbeitet werden kann. Zum anderen bietet der EA auch keine Möglichkeit, die erstellten Analysediagramme automatisiert im Entwurf weiterzuverarbeiten, so dass auch in der Weiterentwicklung keine Zeitersparnis durch die Verwendung des EA in der Analyse erzielt werden kann.

Auch mit Microsoft Powerpoint lassen sich die Analysediagramme erstellen, wobei ein maximaler Freiraum für angepasste Gestaltung bleibt, was bei Modellierungswerkzeugen regelmäßig nicht gegeben ist. Um bei der Modellierung mit Powerpoint zügig voranzukommen und einheitliche Ergebnisse zu erzielen, benötigt man einen ausreichenden Satz von Vorlagen und vorgefertigten Modellierungsobjekten für Powerpoint, wie er in [SA-A-KUHN-10] gegeben ist.



Information 22: Der LDocEditor GUI-Prototyp in Visual Studio .NET 2003

Für die zu erstellenden Schnittstellenprototypen ist es ratsam, diese wo möglich gleich mit der später verwendeten Entwicklungsumgebung zu erstellen, da sie dann im kompletten Prototypen sofort eingesetzt werden können. Diese Vorgehensweise wurde im LDocs-Team im WS 04/05 erfolgreich angewandt.

Zusammenführen der Ergebnisse

Das Zusammenführen der Inhalte aus Powerpoint und Word ist bei C&M explizit im Dokument [C&M-P-KE] geregelt. Das dort beschriebene Vorgehen wurde im Rahmen dieser Studienarbeit durch ein geeignetes *Add-In* für Powerpoint [SA-A-KUHN-04] vereinfacht und die Dokumentation dazu in [SA-A-KUHN-03] und teilweise auch [C&M-P-KE] niedergeschrieben.

Die Schnittstellen, die in der IDE entworfen wurden, lassen sich durch ein Windows *screen capture* (Alt-Druck) leicht als Grafiken in Powerpoint einfügen.

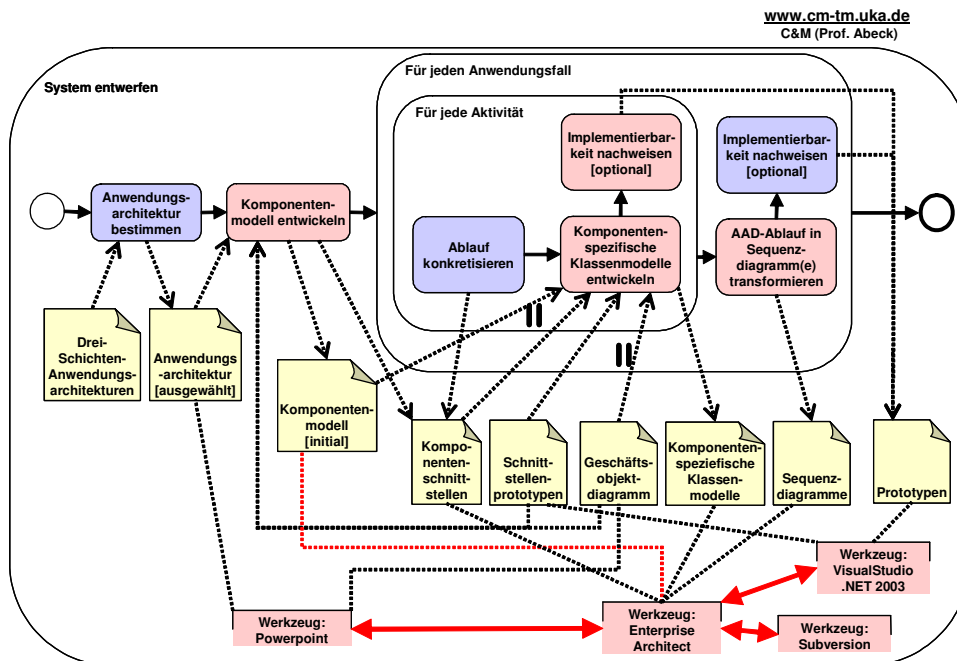
Zusammenwirken der Werkzeuge

Die Produkte der Microsoft Office-Reihe bieten die Möglichkeit, über die Einbettung so genannter OLE-Objekte beinahe jedes Office-Programm als *Plug-In* des anderen verwenden zu können. Diese Fähigkeit wird jedoch von C&M nicht explizit genutzt, z.B. durch direkte Einbettung der Powerpoint-Folien in Word, da sich dahinter eine erhebliche Komplexitätssteigerung gegenüber dem momentan praktizierten Ansatz verbergen würde.

Teamarbeit

Teamarbeit an sich wird durch die Office-Werkzeuge bisher leider nur eingeschränkt unterstützt; es gibt zwar zumindest bei Word die Überarbeiten-Funktionen (Änderungen nachverfolgen), die bei C&M auch regelmäßig genutzt wird, aber bisher kein wirklich taugliches Verfahren zum gemeinsamen Arbeiten an Dokumenten im Sinne der obigen Ausführungen zu diesem Thema. Ein erster guter Schritt in diese Richtung könnten die Microsoft-Produkte Microsoft SharePoint Services und Microsoft SharePoint Portal Server sein. Dies wäre jedoch getrennt zu untersuchen.

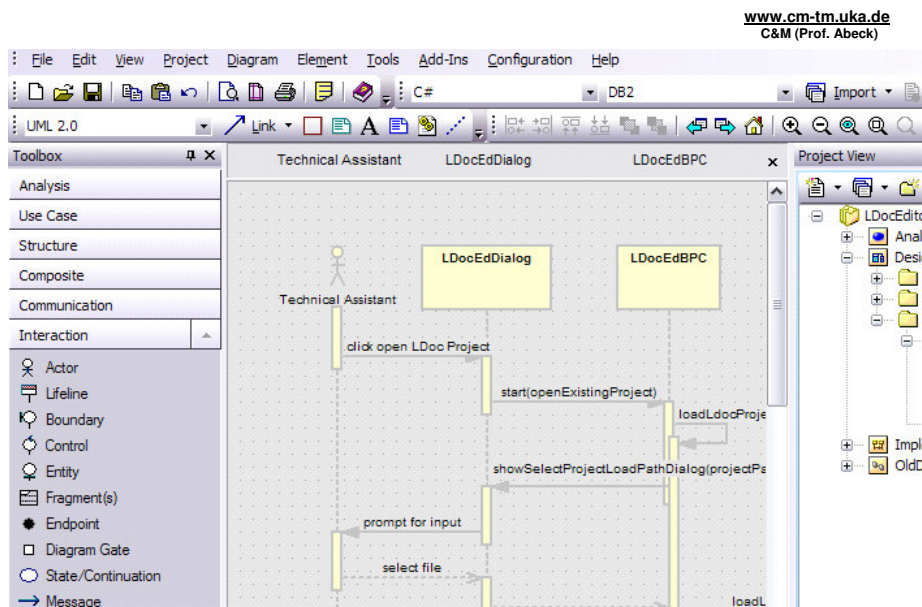
4.2.2 Aktivität „System entwerfen“



Information 23: Aktivität System entwerfen

Powerpoint vs Modellierungswerkzeug

Bei der Softwareentwicklung fällt es leichter graphische Modellierungselemente, wie die der UML, anstelle von textuellen Beschreibungen zu verwenden. Des Weiteren ist es einfacher, sich in ein Softwaresystem einzudenken, wenn man aktiv mit den UML-Diagrammen arbeiten kann. Die in diesem Schritt relevanten Diagramme sind Komponenten-, Klassen- und Sequenzdiagramme, deren Größe für anständiges Arbeiten oftmals die einer Powerpoint-Folie übersteigen, während man bei Modellierungswerkzeugen wie dem EA nicht nur beliebig Platz zum Modellieren hat, sondern auch die nötigen Objekte schnell und einheitlich zeichnen kann und größere Kämpfe für eine einheitliche Formatierung und Anordnung der Modellelemente vermieden werden können. Darüber hinaus arbeitet man beim Modellieren im EA nicht nur mit Diagrammen, sondern kann alle Modellierungsobjekte in einem Projektbaum gruppieren, so dass logisch zusammengehörende Modellteile zusammengefasst werden und schnell wieder aufgefunden werden können. Weitere festgestellte Vorteile der Entwicklung mit dem Enterprise Architect gegenüber Powerpoint ist zum einen, dass man leicht Konsistenz zwischen einzelnen Diagrammen wahren kann. So ändern Methodenaufrufe in Sequenzdiagrammen beispielsweise automatisch ihren Namen wenn der entsprechende Methodenname im Klassendiagramm geändert wird. Zum anderen wird die Unteraktivität Implementierbarkeit nachweisen zumindest auf der Klassenebene durch die Codegenerierungsfunktion des EA unterstützt, so dass hier mit einem kürzeren Zeitaufwand zu rechnen ist. Weitere Vereinfachungen bei der Implementierungsphase sprechen ebenfalls für den Einsatz des Modellierungswerkzeugs. Siehe dazu auch Kapitel 4.2.3. Die rot gestrichelte Verbindung in Information 23 deutet darauf hin, dass es sich auch schon lohnen kann bereits die Komponentenmodelle in EA zu entwerfen, da sie so relativ schnell mit den Klassenmodellen synchronisiert werden können.



Information 24: Sequenzdiagramm im EA

Auch die Nachteile des Enterprise Architect sollen nicht verschwiegen werden: Es ist ein nicht unwesentlicher Einarbeitungsaufwand von Nöten⁴, der für Anwender anderer Modellierungswerkzeuge allerdings merklich geringer ausfällt. Der Einstieg in die Arbeitsweise mit EA wird durch das Dokument [SA-A-KUHN-11] mit einigen Anwendungstipps erleichtert. Ein weiterer Nachteil ist, dass die erstellten Diagramme nicht direkt in der Produktdokumentation vorliegen, sondern in diese erst eingefügt werden müssen, was jedoch außer der Tatsache, dass es sich um einen zusätzlichen Arbeitsschritt handelt, keine weiteren Schwierigkeiten mit sich bringt.

Abgesehen von diesen Vor- und Nachteilen unterstützt der EA auch die Arbeit im Team, was ebenfalls in [SA-A-KUHN-1] dokumentiert ist.

Zusammenführen der Ergebnisse

Theoretisch kann man mit dem EA Reports und Dokumentationen in RTF generieren, die man in das Word-Dokument der Pdok einfügen könnte. Diese Generierung ist auch über zu bearbeitende Vorlagen dem C&M-konformen Kurs angemessen anpassbar. Um dieses zu nutzen, wäre es jedoch erforderlich, in EA auch die entsprechenden Dokumentationstexte einzugeben. Auch würde die Struktur des C&M-konformen Kurses durchbrochen, da die entsprechenden Grafiken dann nicht in den Powerpoint-Dateien vorhanden wären.

Auf der anderen Seite ist es über die Windows-Zwischenablage ziemlich einfach, die Diagramme aus EA in das Powerpoint-Dokument einzufügen und die ergänzenden Informationstexte wie bisher im Word-Dokument zu halten, so dass dies der zu empfehlende Ansatz ist.

Zusammenwirken der Werkzeuge

Für den Enterprise Architect wäre die Interaktion mit Microsoft Office interessant. Dieses Thema wurde im vorigen Absatz kurz erläutert. Die Interaktion mit einem Versionierungssystem wird im nächsten Absatz behandelt und die Interaktion mit Visual Studio im Kapitel 4.2.3.

⁴ Wer an der Universität ein Fach mit Softwareentwicklung studiert, sollte allerdings auch zumindest einmal in der Ausbildung mit einem Modellierungswerkzeug konfrontiert werden.

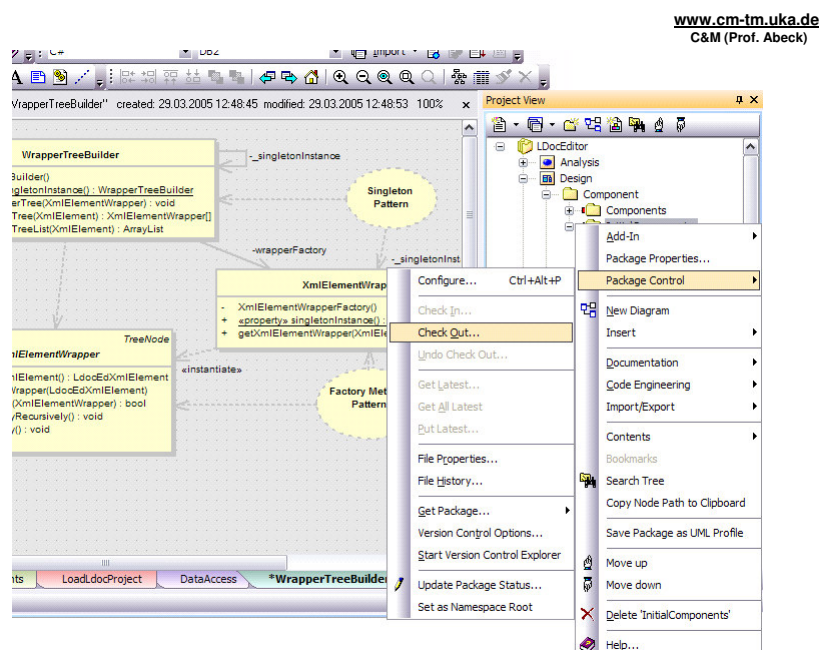
Teamarbeit mit dem Enterprise Architect

Grundsätzlich bietet EA vier verschiedene Möglichkeiten, im Team an einem Projektmodell zu arbeiten:

1. Mittels einer Datenbank (z.B. MySQL), in der das Modell liegt.
2. Das Modell liegt auf einem freigegebenen Dateisystem (z.B. Samba) und kann von mehreren Entwicklern gleichzeitig geöffnet werden.
3. Durch eine spezielle Replikationsfunktion können Ablegerdateien eines Projektes erstellt werden, die nach der Bearbeitung wieder mit dem Originalprojekt synchronisiert werden.
4. Durch Austausch von XMI-Dateien, die Teile des Projektmodells beinhalten.

Bei den ersten beiden Verfahren ist der nötige Synchronisationsaufwand am niedrigsten, die Änderungen am Modell sind für alle sofort verfügbar, jedoch wird zum Arbeiten immer eine aktive Netzverbindung benötigt. Daher sind sie nicht verwendbar, wenn Studenten auch zu Hause ohne Netzverbindung arbeiten können sollen. Mit einer DSL-Internetverbindung und zusätzlichem VPN sind sie gerade so zu gebrauchen, eine bessere Verbindung, wie unmittelbar im C&M-Intranet, wäre wünschenswert.

Auch bei dem dritten Verfahren arbeiten alle Beteiligten an dem gleichen Projektmodell. Hier müssen erstellte Ablegerdateien jedoch von Zeit zu Zeit mit dem Originalprojekt über eine eingebaute Funktion synchronisiert werden, was nicht unaufwändig ist. Darüber hinaus war es nicht ganz nachvollziehbar, wie die Funktion mit kollidierenden Änderungen bei verschiedenen Nutzern umgeht. Es wird zwar eine Kollisionsmanagementfunktion bereitgestellt, diese schien jedoch nicht einwandfrei zu funktionieren und war darüber hinaus sehr unübersichtlich, so dass es nicht ratsam ist, parallel im gleichen Teil des Entwurfsmodells zu arbeiten.



Information 25: Versionskontrolle im EA mit dem SCC SVN proxy

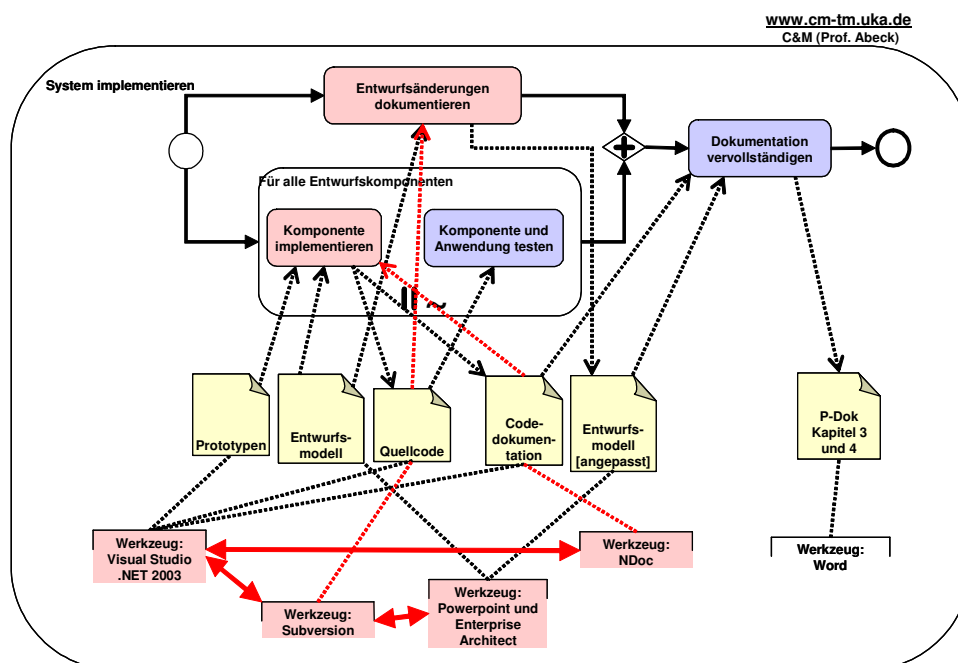
Bei den ersten drei Verfahren arbeiten alle Beteiligten an demselben Objektmodell. Beim vierten kann man zwar auch eine gemeinsame Ursprungsdatei verwenden, doch im Prinzip hat jeder seine eigenen Datei, in die einzelne Pakete aus XMI-Dateien

eingebunden werden. Nachteil dieses Verfahrens ist, dass auch hier die Teilnehmer auf disjunkten Modellteilen arbeiten müssen und es einen etwas erhöhten Aufwand für das Einbinden und Speichern der XMI-Pakete erfordert. Durch geeignete Richtlinien, Konfiguration und Zusatzfunktionen kann dieser Aufwand jedoch relativ gering gehalten werden. Die XMI-Pakete können in der C&M-Verzeichnisstruktur oder vielleicht noch besser in einem Versionierungssystem gespeichert werden. Letzteres wird durch den EA für SCC und CVS sogar direkt unterstützt bzw. mit einem zusätzlichen Proxy auch für Subversion. Ein weiterer Vorteil der vierten Variante ist, dass hier auch unproblematisch verschiedene Versionen des EA an einem Projekt arbeiten können, was bei den anderen nicht möglich ist.

Aus diesen Überlegungen heraus wurde in [SA-A-KUHN-11] die vierte Vorgehensweise beschrieben.

4.2.3 Aktivität „System implementieren“

Diese Aktivität teilt sich in einige Unteraktivitäten auf, die sich auf unterschiedliche Weise mit den zu untersuchenden Werkzeugen unterstützen lassen.



Information 26: Aktivität System implementieren

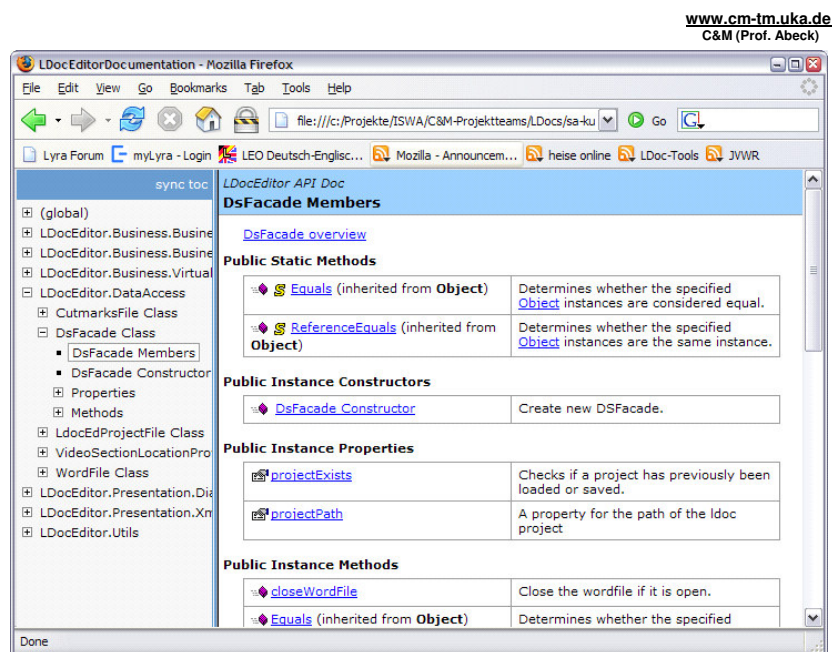
Bei der Aktivität Komponente Implementieren können die Klassenrumpfe wie schon in Kapitel 4.2.2 erwähnt aus dem Klassenmodell in Enterprise Architect generiert werden. Die eigentliche Arbeit am Quellcode wird mit Visual Studio durchgeführt. Auf eine detailgenaue Beschreibung der Arbeit mit Visual Studio wurde verzichtet. Und zwar 1. aus zeitlichen Gründen, 2. weil Visual Studio im Wesentlichen zu bedienen ist wie die meisten anderen IDEs auch und 3. stand die eigentlich zu untersuchende Version von Visual Studio .NET 2005 erst ganz gegen Ende der Studienarbeit zur Verfügung. In [SA-A-KUHN-13] sind ein paar Besonderheiten im Umgang mit Visual Studio .NET 2003 beschrieben.

Im Diagramm in Information 26 ist die Codedokumentation über einen roten Pfeil rückwärts verknüpft mit der Aktivität. Dadurch soll zum Ausdruck kommen, dass man bei mehreren aufeinander folgenden Implementierungsschritten von der Dokumentation der jeweils vorhergehenden Schritte profitieren kann.

Auch die Aktivität Entwurfsänderungen dokumentieren hat eine eingehende Information hinzubekommen: Durch die Funktion *reverse engineering* bzw. der Quellcode-Modell-Synchronisation des EA kann man die Klassenmodelle teilautomatisiert direkt vom Quellcode anpassen.

Zusammenführen der Ergebnisse

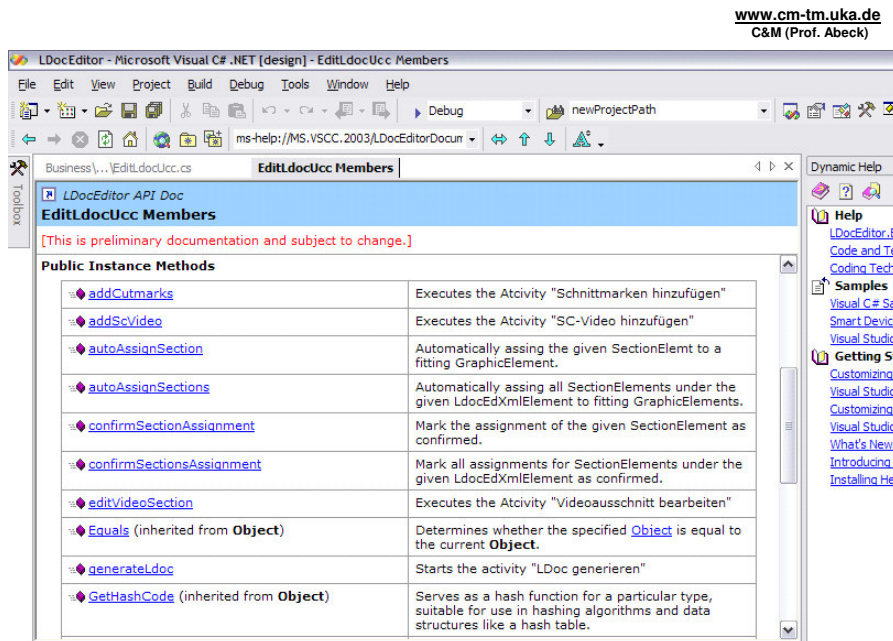
Im Schritt Dokumentation vervollständigen sollten die Implementierungsdokumentation und das angepasste Entwurfsmodell einfließen. Letzteres wird wie in 4.2.3 bewerkstelligt. Die Idee, eine generierte Implementierungsdokumentation (möglicherweise automatisiert) in die Pdok zu integrieren, hat sich bei einigen Tests als nicht sehr praktikabel erwiesen, da dabei selbst bei einem kleineren Projekt wie dem LDocEditor schnell über 100 Word-Seiten generiert werden, die in der Produktdokumentation nicht wirklich von Interesse sind. Stattdessen kann man eine API-Dokumentation des Produktes zusätzlich zur Pdok bereitstellen und nur die wirklich wichtigen Anmerkungen aus dieser in die Pdok hineinkopieren, bzw. in der Pdok im Implementierungsteil zusätzliche Informationen liefern. Eine API-Dokumentation in verschiedenen Formaten kann aus der Quellcodedokumentation mit Hilfe des Werkzeugs NDoc generiert werden.



Information 27: LDocEditor API-Doc in Firefox

Zusammenwirken der Werkzeuge

Für die erzeugte API-Dokumentation besteht sogar die Möglichkeit, diese wieder so in Visual Studio einzubinden, dass sie dort wie die normale .NET-Framework-Dokumentation erscheint und zu verwenden ist und so einen direkten Mehrwert bei der Weiterentwicklung des Produktes liefert. Dies ist ebenso wie die einfache Generierung der Dokumentation mit NDoc in [SA-A-KUHN-14] beschrieben.

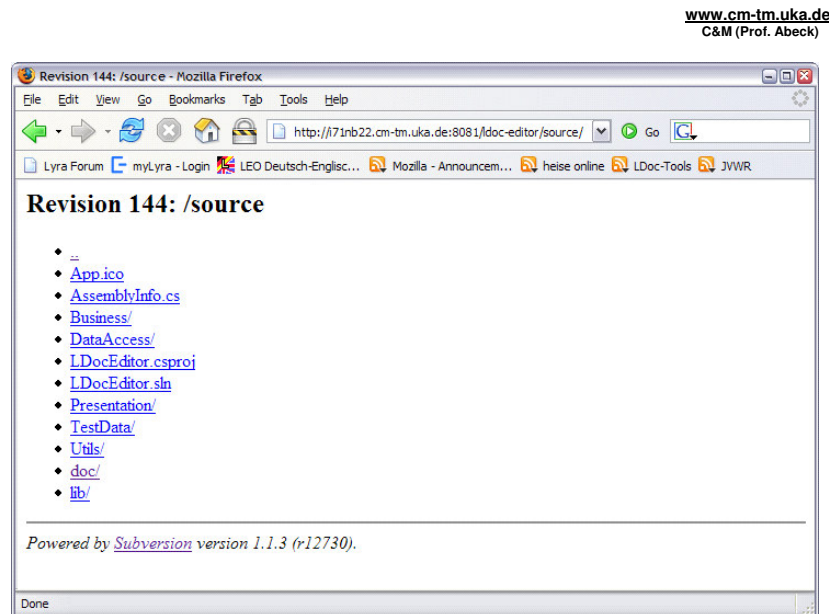


Information 28: LDocEditor API-Doc in Visual Studio

Die Zusammenarbeit zwischen Subversion und Enterprise Architect wurde bereits in 4.2.2 erläutert und diejenige zwischen Subversion und Visual Studio wird im nächsten Kapitel vorgestellt.

Teamarbeit

Die Implementierung im Team läuft oftmals so ab, dass verschiedene Teammitglieder in unterschiedlichen Bereichen am Quellcode arbeiten. Dabei werden jedoch meist mehrere Dateien an verschiedenen Stellen bearbeitet. Darüber hinaus verwendet jeder, der zu Hause implementiert, seine eigene Kopie der Quellcodes. Arbeitet man hier ohne Werkzeugunterstützung, so kommt es schnell zu Problemen beim Zusammenführen der beiden bearbeiteten Kopien: Um die Zusammenführung effizient zu gestalten, müsste sich jeder Beteiligte sich genau notiert haben, in welchen Dateien er welche Änderungen vorgenommen hat. Bei zwei Beteiligten wäre das noch praktikabel, bei mehreren Beteiligten stößt man dabei jedoch bald an eine Machbarkeitsgrenze.



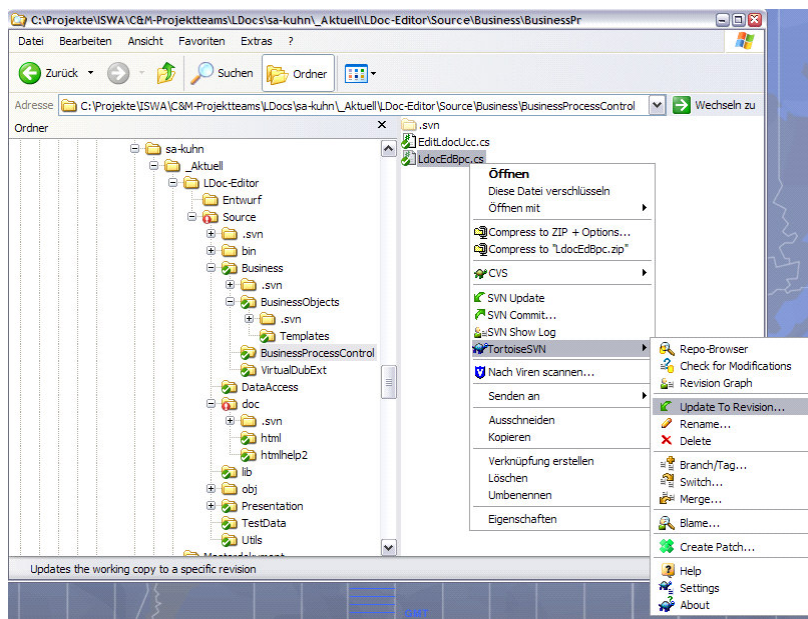
Information 29: WebDav-Ansicht im Firefox-Browser

Im LDocs-Team wurde zur Erstellung des LDocEditor-Prototyps das Versionierungssystem Subversion, der selbsternannte Nachfolger von CVS, getestet, welches gegenüber dem älteren System tatsächlich einige Verbesserungen aufweist. Subversion ist als Modul für den Apache-Webserver⁵ verfügbar und basiert auf dem WebDav-Protokoll. Es hat die von CVS bekannten *features* wie Historisierung und *branching* und erlaubt es, gleichzeitig an den Quellen zu arbeiten. Dabei wird von der Annahme ausgegangen, dass die Fälle, an denen mehrere Personen die gleiche Datei ändern, selten auftreten, und daher werden bearbeitete Dateien für andere Teammitglieder nicht gesperrt. Sollte doch einmal eine Kollision auftreten, müssen die beiden Änderungssätze der Datei zusammengeführt werden, was von den Subversion-Clients aber gut unterstützt wird. Die Grundprinzipien und Arbeitsweise von Subversion, die Installation und Inbetriebnahme des SVN-Servers, sowie die verschiedenen Client-Befehle sind im Buch [CF+04] und der Subversion-Hilfe ausführlich erläutert.

Bei der Erstellung des LDocEditors hat das Subversion-*Repository* gute Dienste geleistet. Es war für alle Beteiligten einfach, immer an den aktuellsten Quellen zu arbeiten. Die einzige Schwierigkeit, die sich bei der Verwendung ergab, war die Umbenennung einzelner Dateien. Dies wird jedoch von Subversion im Gegensatz zu CVS unterstützt, so dass es dabei auch keine Probleme gab. Getestet wurden für diese Studienarbeit zwei verschiedene Clients.

⁵ Das bedeutet auch dass Subversion alle Authentifizierungsmodi des Apache-Webservers verwenden kann. Insbesondere und besonders interessant für C&M auch die Möglichkeit mit einem geeigneten Apache-Modul gegen eine NT-Domäne zu prüfen.

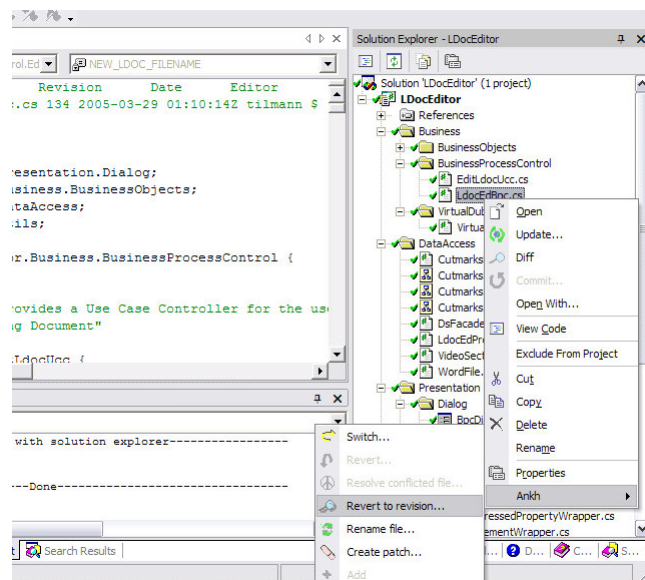
www.cm-tm.uka.de
C&M (Prof. Abeck)



Information 30: TortoiseSVN

Der TortoiseSVN ist ein Client, der sich in den Windows-Explorer integriert und über das Kontextmenü von Dateien und Verzeichnissen bedient wird. Er unterstützt fast den vollständigen SVN-Befehlsumfang; in keinem Fall war ein Funktionsaufruf nötig, der von TortoiseSVN nicht bereitgestellt wurde.

www.cm-tm.uka.de
C&M (Prof. Abeck)

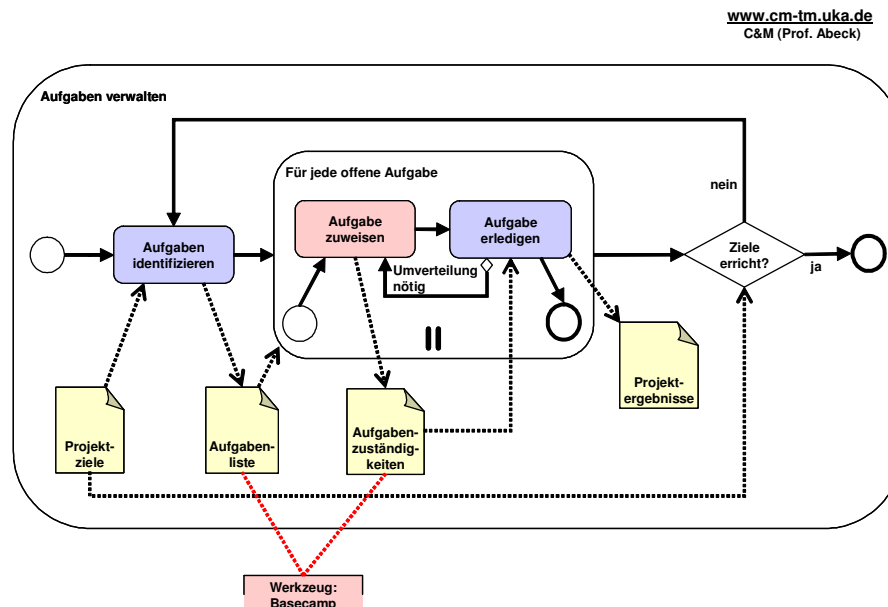


Information 31: AnkhSVN

Darüber hinaus wurde das *SVN-Plugin* für Visual Studio 2003 mit dem Namen AnkhSVN verwendet. Dieses bietet nur einen geringeren Funktionsumfang und sollte richtig konfiguriert werden (z.B. mit der Konfigurationsdatei [SA-A-KUHN-17]). Es funktioniert dann aber auch einwandfrei, so dass das Zusammenwirken von Visual Studio und SVN keine Schwierigkeiten bereitet. In [SA-A-KUHN-16] befindet sich eine Kopie

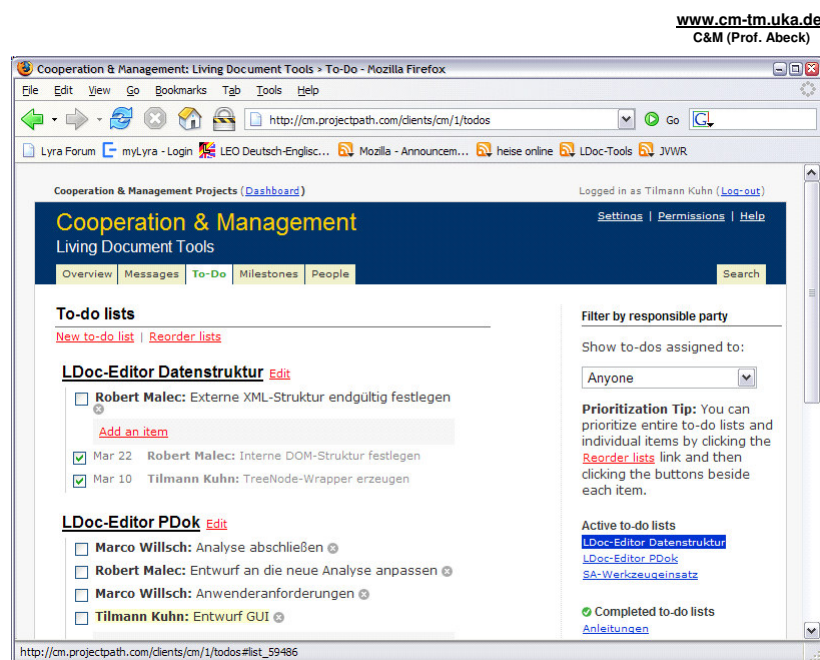
der eingesetzten client-übergreifenden Subversion-Konfigurationsdatei, welche Auswirkungen auf das Verhalten aller Clients herbeiführt.

4.2.4 Aktivität „Aufgaben verwalten“



Information 32: Aktivität Aufgaben verwalten

Diese Aktivität wird bei C&M von jedem Team selbstständig durchgeführt. Im LDocs-Team wurde dazu, wie in Kapitel 2.5.2 angedeutet, kurzentschlossen während des Semesters ein wenig mit dem *Hosted Service* Basecamp experimentiert, um die Aufgabenverteilung transparenter zu gestalten als bisher. Bisher wurden die Aufgaben aufgrund von Einzelabgesprächen zwischen den Teammitgliedern aufgeteilt.

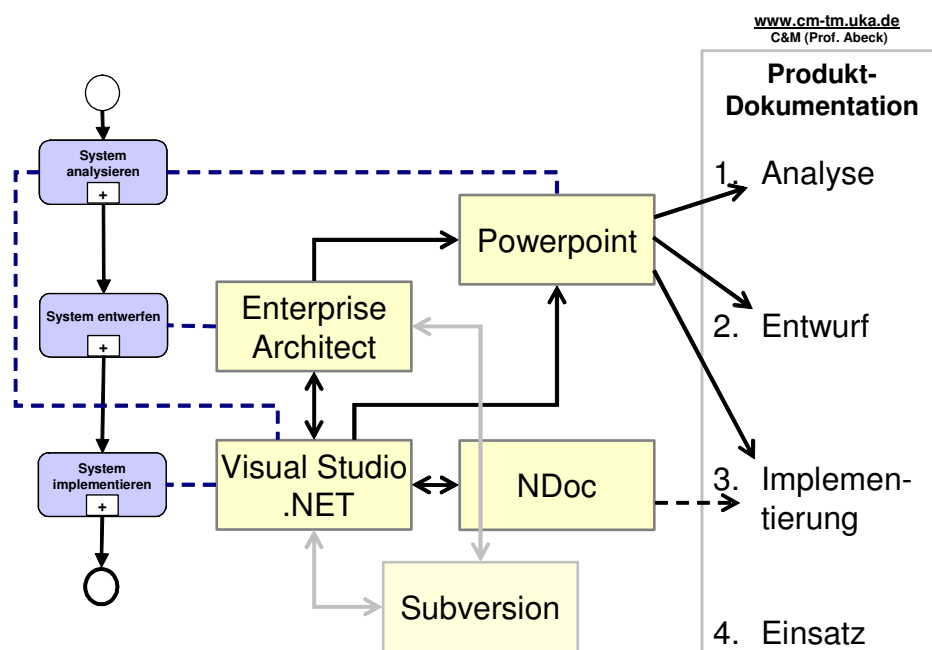


Information 33: *Todo*-Listenverwaltung in Basecamp

Basecamp ist ein webbasiertes System, das sehr einfach gehalten ist, also nicht die Komplexität einer teuren Projektmanagementlösung bietet, sondern hauptsächlich Wert auf leichte Bedienbarkeit legt. So gelang es den Teammitgliedern auch, im Wesentlichen ohne Einarbeitung das Werkzeug zu verstehen und nutzen zu können. Man kann mit Basecamp *Todo*-Listen verwalten, wie in Information 33 dargestellt; dabei wird man per E-Mail und RDF über Änderungen an der Liste informiert. Nachteil von Basecamp im Speziellen ist, dass man keine Schnittstelle zur C&M-Mitgliederverwaltung hat, sondern die einzelnen Mitarbeiter per Hand in das Werkzeug eintragen muss, was schätzungsweise einen Arbeitsaufwand von ca. 0,5 h pro Semester und Team ausmachen würde. Die Reaktionen der studentischen Teammitglieder waren sehr unterschiedlich: Zwei fanden das Werkzeug nützlich, ein Teammitglied meinte sogar: „Genau so etwas hat mir bei C&M noch gefehlt.“ Ein weiteres weigerte sich schlichtweg (ohne Angabe von Gründen) Basecamp zu benutzen.

5 ZUSAMMENFASSUNG UND AUSBLICK

Für die Softwareentwicklung gibt es eine riesige Menge an Werkzeugangeboten, bei denen die Qualität ebenso weit gestreut ist wie der Lizenzpreis. Namhafte Unternehmen verwenden vorrangig ihre eigenen Produkte zur Softwareentwicklung, wahrscheinlich nicht zuletzt um Kosten zu sparen. Über kleinere Unternehmen sind kaum Angaben bekannt. Generell werden in der Softwareentwicklung für Dokumente oft Microsoft-Office-Produkte eingesetzt. Ein geeignetes CASE-Werkzeug und ein Source-Management/ Versionierungssystem fehlte bei keinem Teilnehmer der Umfrage. Aus einer vorbewerteten Aufstellung wurden die preislich günstigen Werkzeuge Powerpoint, Enterprise Architect, Visual Studio .NET 2005, NDoc und Subversion zum testweisen Einsatz bei C&M ausgewählt. Eine Möglichkeit, diese Werkzeuge zu verwenden wurde in Kapitel 4 aufgezeigt und ist hier noch einmal zusammengefasst:



Information 34: Phasen, Werkzeuge und die Produktdokumentation

Die gestrichelten Verbindungen links ordnen die Werkzeuge grob den einzelnen Aktivitäten zu, schwarze und graue Pfeile sollen den Informationsfluss darstellen.

In einem ersten Schritt System analysieren arbeitet man am besten kundenfreundlich mit Powerpoint, das auch in den Sitzungen verwendet wird. Für die Erstellung von GUI-Prototypen empfiehlt sich zwecks Weiterverwendung, diese in der später verwendeten Entwicklungsumgebung, hier Visual Studio.NET, zu erstellen und *Screen Shots* über Powerpoint-Folien in die Pdok einzubringen.

Im Entwurf kann man Zeit sparen, wenn man auf das spezielle UML-Modellierungswerkzeug Enterprise Architect zurückgreift, es ist jedoch ein gewisser Einarbeitungsaufwand vorhanden. Erzeugte Diagramme können leicht in Powerpoint-Folien eingefügt werden.

In der Implementierungsphase komplettiert man den anfänglich von EA generierten Quellcode mit Visual Studio. Änderungen an den Quelldateien können von EA zumindest teilweise eingelesen und in das UML-Modell übernommen werden, was die Anpassung der Entwurfsdokumentation vereinfacht.

Mit NDoc kann man die Codekommentare in ein ansehnliches Format konvertieren und diese auch wieder in VS integrieren, so dass sie fortfolgenden Implementierungsphasen zugute kommen. Eventuell kann man auch wichtige Teile der generierten API-Dokumentation in die Implementierungsdokumentation einfließen lassen.

Zu guter Letzt kann das Versionierungssystem Subversion als Daten-*Backend* sowohl für Enterprise Architect als auch für den Quellcode genutzt werden, so dass in der Implementierungsphase lästiges Kopieren und Zusammenstückeln von Quellcodeteilen weitestgehend entfällt.

Ein interessanter Punkt, der zukünftig noch zu untersucht werden könnten, ist, ob die neue Microsoft Technologie der SharePoint Services bzw. SharePoint Portal Server in einer neuen Version, die Zusammenarbeit auf den Office-Dokumenten der Produktdokumentation effizienter zulässt. Das manuelle Zusammenführen verschiedener Dokumentversionen ist noch ein Manko der Teamarbeit an der Pdok.

Sicherlich auch wird noch interessant werden, wie die Ergebnisse der aktuellen Diplomarbeit Qualitätssicherung bei der Softwareentwicklung am Beispiel der LDoc-Tools von Marco Willsch den Softwareentwicklungsprozess weiterentwickeln, welche zusätzlichen Werkzeuge für die Softwaretests und die Dokumentierung der Testergebnisse verwendet werden können und wie diese in das in dieser Arbeit vorgestellte Gefüge der Werkzeuge zu integrieren sind.

In direkter Ergänzung zu dieser Arbeit sollte noch Visual Studio .NET 2005 daraufhin untersucht werden, wie tauglich es zur UML-Modellierung ist. Aktuelle Informationen und *Screen Shots* deuten allerdings darauf hin, dass es lediglich Fähigkeiten zur Erstellung von Geschäftsprozess- und Klassendiagrammen bieten wird, die zudem eher ein Microsoft-spezifisches Aussehen haben als das der UML- oder BPMN-Standards.

6 VERZEICHNISSE

6.1 Bearbeitete Dokumente

Hier werden die im Rahmen dieser Studienarbeit bearbeiteten bzw. erstellten Dokumente aufgelistet. Diejenigen unter ihnen, die nicht im C&M-Literaturverzeichnis stehen, wurden mit SA-A-KUHN-?? für Studienarbeit-Anlage-Kuhn-<Nummmer> bezeichnet. Alle Dokumente wurden mit dieser Arbeit auf CD ausgeliefert und sind in der Tabelle verlinkt. In der rechten Tabellenspalte findet sich der Name des Dokumentes sowie eine kurze Beschreibung welche Beiträge darin geleistet wurden.

- | | |
|----------------|---|
| [C&M-P-KE] | Cooperation & Management, C&M-KURSENTWICKLUNG,
http://www.cm-tm.uka.de/extern , Ordner "Profil", Universität
Karlsruhe (TH), C&M (Prof. Abeck).
C&M-P-KE_cm-kursentwicklung_04-12-16-kuhn.doc
C&M-P-KE_cm-kursentwicklung_04-12-16-kuhn.ppt
Beiträge:
Änderungen in Kapitel 2.4.1
Kapitel 2.4.2 |
| [SA-A-KUHN-01] | Der Fragebogen zu Werkzeugen in der Softwareentwicklung, der
an mehrere Unternehmen versandt wurde und die Antworten.
SA-A-KUHN-01_fragebogen-werkzeugeinsatz_04-11-18-kuhn.doc
Antworten\Fragebogen_Werkzeugeinsatz_CAS.doc
Antworten\Fragebogen_Werkzeugeinsatz_HP.txt
Antworten\Fragebogen_Werkzeugeinsatz_IBM_1.doc
Antworten\Fragebogen_Werkzeugeinsatz_IBM_2.doc
Antworten\Fragebogen_Werkzeugeinsatz_SAP.txt
Beiträge:
Komplettes Dokument |
| [SA-A-KUHN-02] | Die vorige Produktdokumentation der LivingDocument-Tools
SA-A-KUHN-02_P-Dok_ldoc-tools_04-12-21-kuhn.doc
SA-A-KUHN-02_P-Dok_ldoc-tools_04-11-29-kuhn.ppt
Beiträge:
Umformatiert und Änderungen für eine Modularisierung der
LDoc-Tools |
| [SA-A-KUHN-03] | Die Produktdokumentation des LDoc-C&M CourseBuilders
SA-A-KUHN-03_P-Dok_ldoc-c&mcoursebuilder_05-04-20-kuhn.doc
SA-A-KUHN-03_P-Dok_ldoc-c&mcoursebuilder_05-04-20-kuhn.ppt
Beiträge:
Kapitel 2, 3 und 4, teilweise Kapitel 1 |
| [SA-A-KUHN-04] | Das LDoc-C&M CourseBuilder-Add-In
SA-A-KUHN-04_ldoc-c&mcoursebuilder-addin-office11_05-04-20-kuhn.ppt
Beiträge:
Komplettes Add-In |
| [SA-A-KUHN-05] | Die Anleitung zur Nutzung des LDoc-Generators |

[SA-A-KUHN-05_anleitung-ldoc-generator_05-01-21-kuhn.doc](#)
[SA-A-KUHN-05_anleitung-ldoc-generator_05-01-21-kuhn.ppt](#)

Beiträge:

Umformatiert, korrigiert und aktualisiert.

- [SA-A-KUHN-06] Die Anleitung zur Vorlesungsaufzeichnung
[SA-A-KUHN-06_anleitung-vorlesungsaufzeichnung_05-02-07-kuhn.doc](#)
[SA-A-KUHN-06_anleitung-vorlesungsaufzeichnung_05-01-21-kuhn.ppt](#)

Beiträge:

Komplettes Dokument

- [SA-A-KUHN-07] Die Anleitung zum Tablet-PC-Anschluss im Hörsaal -102
[SA-A-KUHN-07_anleitung-tablet-pc-anschluss-102_05-02-16-kuhn.doc](#)
[SA-A-KUHN-07_anleitung-tablet-pc-anschluss-102_05-02-16-kuhn.ppt](#)

Beiträge:

Komplettes Dokument

- [SA-A-KUHN-08] Die Anleitung zum Tablet-PC-Anschluss im Hörsaal Forum A+B
[SA-A-KUHN-08_anleitung-tablet-pc-anschluss-forum_05-02-16-kuhn.doc](#)
[SA-A-KUHN-08_anleitung-tablet-pc-anschluss-forum_05-02-16-kuhn.ppt](#)

Beiträge:

Komplettes Dokument

- [SA-A-KUHN-09] Die Anleitung zur Nutzung der Living Documents
[SA-A-KUHN-09_anleitung-ldoc-nutzung_05-01-28-kuhn.doc](#)
[SA-A-KUHN-09_anleitung-ldoc-nutzung_05-01-21-kuhn.ppt](#)

Beiträge:

Komplettes Dokument

- [SA-A-KUHN-10] Vorlage mit Modellierungsobjekten für UML- und BPMN-Modellierung unter Powerpoint.
[Technologien_Werkzeuge\Powerpoint\vorlage_modellierungsobjekte_05-03-10-kuhn.ppt](#)

Beiträge:

Komplettes Dokument

- [SA-A-KUHN-11] Anwendungstipps Enterprise Architect
[Technologien_Werkzeuge\EnterpriseArchitect\anwendungstipps_enterprise-architect_05-04-05-kuhn.doc](#)
[Technologien_Werkzeuge\EnterpriseArchitect\anwendungstipps_enterprise-architect_05-04-03-kuhn.ppt](#)

Beiträge:

Komplettes Dokument

- [SA-A-KUHN-12] Enterprise Architect Beispielprojekt
Für EA 3.6 [EA\Testprojekt_36\LDocEditor_ea36_05-04-02-kuhn.eap](#)
Für EA 4.5 [EA\Testprojekt\LDocEditor_ea45_05-04-02-kuhn.eap](#)

Beiträge:

Teile des Objektmodells, Aufbereitung des Projekts.

- [SA-A-KUHN-13] Anwendungstipps Visual Studio 2003
[Technologien_Werkzeuge\VisualStudio2003\anwendungstipps_visualstudio2003_05-04-02-kuhn.doc](#)
[Technologien_Werkzeuge\VisualStudio2003\anwendungstipps_visualstudio2003_05-04-02-kuhn.ppt](#)
Beiträge:
Komplettes Dokument
- [SA-A-KUHN-14] Anwendungstipps NDoc
[Technologien_Werkzeuge\NDoc\anwendungstipps_ndoc_05-04-03-kuhn.doc](#)
[Technologien_Werkzeuge\NDoc\anwendungstipps_ndoc_05-04-03-kuhn.ppt](#)
Beiträge:
Komplettes Dokument
- [SA-A-KUHN-15] NDoc Beispielprojekt
Projektdatei: [LDocEditor\LDocEditor.ndoc](#)
Ergebnisdokumente: [LDocEditor\doc](#)
Beiträge:
Komplettes Projekt
- [SA-A-KUHN-16] Subversion Client-Konfigurationsdatei
[SA-A-KUHN-16_subversion-config_05-04-01-kuhn.txt](#)
Beiträge:
Einige geänderte Einstellungen zur Originaldatei
- [SA-A-KUHN-17] AnkhSVN-Konfigurationsdatei
[SA-A-KUHN-17_ankhsvn-config_05-04-01-kuhn.xml](#)
Beiträge:
Eine geänderte Einstellung zur Originaldatei
- [SA-A-KUHN-18] Die Produktdokumentation des LDoc-Editors
[SA-A-KUHN-18_pdoc_ldoc-editor_05-04-02-kuhn.doc](#)
[SA-A-KUHN-18_pdoc_ldoc-editor_05-04-02-kuhn.ppt](#)
Beiträge:
Kapitel 3.5.1 und 3.5.2
- [SA-A-KUHN-19] Visual Studio 2003 Beispielprojekt mit SVN-Sandbox und AnkhSVN
[LDocEditor\LDocEditor.sln](#)
- [SA-A-KUHN-20] Test der Word-Masterdokument-Funktion zur Erstellung von Dokumenten, die mehrere C&M-konforme Kurse enthalten.
[Masterdokument](#)
- [SA-A-KUHN-21] Das Stundenkonto das im Zuge dieser Studienarbeit entstand
[SA-A-KUHN-21_kuhn-stundenkonto-kw13.xls](#)
- [SA-A-KUHN-22] Die aus dem LDoc-Editor-Code mit NDoc generierte Dokumentation
[LDocEditor\doc](#)

6.2 Untersuchte Werkzeuge

Enterprise Architect	- Enterprise Architect, Version 4.51.749 EnterpriseArchitect\easeup.exe - MDG link for Visual Studio.NET EnterpriseArchitect\VSBridge.exe
Subversion	- Subversion, Version 1.1.3 Subversion\svn-1.1.3-setup.exe - TortoiseSVN, Version 1.1.3 Subversion\TortoiseSVN-1.1.3-UNICODE svn-1.1.3.msi - AnkhSVN, Version 0.5.5 Subversion\AnkhSetup-0.5.5.1653.msi - PushOk, SVN SCC proxy, Version 1.1.3 Subversion\svnproxy.exe
NDoc	- NDoc, Version 1.3.1 NDoc\NDoc-v1.3.1.msi - Visual Studio .NET 2003 Help Integration Kit NDoc\VSHIK2003.exe - H2Reg, Version 1.36 NDoc\h2reg_setup136.exe
Visual Studio.NET	Visual Studio.NET 2003, Version 7.1
Word, Powerpoint	Microsoft Office 2003 SP 1

6.3 Abkürzungen und Glossar

Abkürzung oder Begriff	Langbezeichnung und/oder Begriffserklärung
API	<i>Application Programming Interface</i> Definierte Stelle, an der eine Anwendung festgelegte Funktionen (z.B. zur Datenübertragung, Transaktionskontrolle, Namensauflösung) in Anspruch nehmen kann.
BPM	<i>Business Process Management</i> Prozessmodellierung oder <i>Business Process Modeling</i> umfasst den Entwurf und die Implementierung von Geschäftsprozessen mit Hilfe eines BPM-Systems. Dazu stehen in der Regel graphische Editoren zur Verfügung. Um die Verbindung zu den vorhandenen Applikationen und Komponenten herstellen zu können, werden die entsprechenden Spezifikationen in das System importiert oder dort definiert, soweit keine geeigneten Importmöglichkeiten bestehen. Die fertigen Prozessdefinitionen können direkt von einer integrierten <i>Business Process Engine</i> verarbeitet werden oder lassen sich in einer Form exportieren, die dann von einer Integrationsplattform bzw. der dort integrierten <i>Process Engine</i> importiert werden kann.

	Eine wichtige Spezifikation ist die <i>Business Process Modeling Notation</i> (BPMN) der <i>Business Process Management Initiative</i> (BPMI). [Wikipedia, http://de.wikipedia.org/wiki/Business_Process_Management]
C&M	Cooperation & Management Name der an der Universität Karlsruhe (TH) angesiedelten Forschungsgruppe.
C&M-T-Map	C&M-Technologies&Tools-Map In der Forschungsgruppe C&M verwendete Landkarte, in der alle Technologien und (Entwicklungs-) Werkzeuge in strukturierter Form aufgeführt sind, die in Projekten eingesetzt werden und zu denen Kompetenzen in der Forschungsgruppe vorhanden sind.
EA	Akronym für Enterprise Architect
ERP	<i>Enterprise Resource Planning</i> → Unternehmenssoftware
Geschäftsprozess	Ein Geschäftsprozess ist eine Abfolge von Aktivitäten, die der Erzeugung eines Produktes oder einer Dienstleistung dienen. Er wird durch ein oder mehrere Ereignisse gestartet und durch ein oder mehrere Ereignisse abgeschlossen. Es liegt eine Organisationsstruktur zu Grunde.
GUI	<i>Graphical User Interface</i> Grafische Benutzeroberfläche
Komponente	Ein UML-Modellelement, das eine ausführbare und austauschbare Softwareeinheit mit definierten Schnittstellen und eigener Identität beschreibt. Es sind Komponentendefinitionen (z.B. "Lehrveranstaltung") und Komponenteninstanzen (z.B. "INFORMATIK I") zu unterscheiden.
LDoc	Living Document Ein in Standard-HTML-Format vorliegendes Textdokument (Document) mit eingebundenen Grafiken, die durch Anklicken um (i.d.R. multimediales) Material ergänzt werden können und damit zum Leben (Living) erweckt werden.
MS	Microsoft
Pdok	Produktdokumentation Ein gemäß einer vorgegebenen Struktur aufgebautes Dokument, das im Rahmen der bei C&M eingeführten strukturierten Softwareentwicklung erstellt wird und zur Dokumentation eines C&M-Softwareprodukts dient.
Profildokumente	Dokumente, durch die verschiedene das Profil der Forschungsgruppe

C&M bestimmende Aspekte beschrieben werden. Insofern stellen diese über www.cm-tm.uka.de/extern über das Web zugreifbaren Dokumente die wichtigsten Team-Dokumente von C&M dar.
Beispiele: KOMPETENZEN, ARBEITSWEISE UND ORGANISATION, ENTWICKLUNG VON KURSEN, ENTWICKLUNG VON SOFTWARE (vorliegendes Dokument)

RTF	<i>Rich Text Format</i> Austauschformat für formatierten Text zwischen verschiedenen Anwendungen.
SVN	Abkürzung für Subversion
System	Eine Menge von Elementen, die untereinander in einer wohl definierten Beziehung stehen.
UML	<i>Unified Modeling Language</i> Sprache, die aus graphischen Elementen besteht und zur semi-formalen Beschreibung von beliebigen Gegenständen (z.B. Software-Systeme oder Geschäftsbereiche) geeignet ist. Language to describe any objects (e.g., software systems or business units) in a semi-formal way using graphical elements.
Unternehmenssoftware	Unternehmenssoftware umfasst jede betriebswirtschaftliche Software, die in einem Unternehmen zur Bewältigung des Geschäftsbetriebes eingesetzt wird. Ihre Aufgabe ist die Abbildung der realen Welt des Unternehmens in Form von Datenmodellen. Den Mitarbeitern des Unternehmens dient sie zur Information, Verwaltung, Organisation, Planung. Als englischer Begriff wird häufig synonym <i>Enterprise Ressource Planning (ERP)</i> verwendet, obwohl dieser häufig mit einem engeren Fokus verbunden ist. Als Beispiel für Unternehmenssoftware kann SAP R/3 mit den darauf aufsetzenden Modulen angeführt werden. Im allgemeinen nicht unter Unternehmenssoftware gefasst werden Produkte wie Textverarbeitungen, allgemeine Tabellenkalkulationen usw. Englisch: Enterprise Resource Planning (ERP) Beispiel: SAP R/3 „Unternehmenssoftware umfasst alle betriebswirtschaftliche Software, die ein Unternehmen zur Bewältigung seines Geschäftsbetriebes einsetzt. Sie bildet die reale Welt des Unternehmens als Datenmodell ab. Den Mitarbeitern des Unternehmens dient sie zur Information, Verwaltung, Organisation, Planung [Wikipedia, http://de.wikipedia.org/wiki/Unternehmenssoftware].
URL	<i>Uniform Resource Locator</i> Eine Zeichenkette, die eine Ressource, wie z.B. eine Web-Seite, eindeutig im World Wide Web identifiziert.

VS	Akronym für Visual Studio
WWW	World Wide Web Wichtigste Anwendung des Internet.
XMI	<i>XML Metadata Interchange</i> Austauschformat für Modelle zwischen verschiedenen Modellierungswerkzeugen.

6.4 Index

C&M <i>Siehe</i> Cooperation & Management	Programmierung .. 11 , 12, 13, 15, 18, 21
C&M-T-Map..... 8, 33, 54	Projektmanagement10, 12, 13, 14, 15, 16, 20
Cooperation & Management..... 3, 7, 54	Recherche..... 8, 14 , 15, 16
Dokumentation 11 , 16, 18, 19, 22, 39, 41, 42, 43, 49	Softwareentwicklungsprozess7, 8, 9, 11, 15, 19, 23 , 33 , 49
EA <i>Siehe</i> Enterprise Architect	Source-Management 11 , 12, 13, 16, 18, 22, 48
Enterprise Architect8, 11, 15, 18, 21, 22, 36, 38, 39, 40, 41, 42, 43, 48, 49, 51, 53	Subversion8, 9, 11, 16, 18, 22, 41, 43, 44, 45, 48, 49, 52, 53
Frameworks 11 , 12, 13, 15	SVN..... <i>Siehe</i> Subversion
Kategorien..... 8, 10 , 15, 19, 20, 33	Teamkommunikation10, 12, 13, 14, 15, 17, 20
Modellierung 11 , 12, 13, 14, 15, 18, 21, 34, 36, 38	Test..... 11 , 12, 13, 15, 49
NDoc8, 16, 19, 22, 42, 43, 48, 49, 52, 53	Umfrage 8, 11 , 14, 16, 48
Pdok <i>Siehe</i> Produktdokument	Versionierung 11 , 12, 13, 14, 16, 18, 21, 22, 39, 41, 44, 48, 49
Powerpoint8, 11, 13, 18, 19, 21, 22, 33, 36, 37, 38, 48, 51, 53	Visual Studio8, 11, 15, 18, 21, 22, 39, 41, 43, 45, 48, 49, 52, 53
Produktdokumentation3, 7, 8, 24, 25, 33, 39, 42, 48, 49, 50, 52, 54	VS <i>Siehe</i> Visual Studio

6.5 Informationen

Information 1: EVALUATION VON WERKZEUGUNTERSTÜTZUNG BEI DER SOFTWAREENTWICKLUNG.....	5
Information 2: Software-Kategorien und Dokumente.....	10
Information 3: Geschäftsprozess Softwareentwicklung bei C&M.....	24
Information 4: Unterprozess System modellieren BPMN-Diagramm.....	25
Information 5: Unterprozess System analysieren Beschreibung	26
Information 6: Unterprozess System analysieren BPMN-Diagramm.....	26
Information 7: Unterprozess System entwerfen Beschreibung.....	27
Information 8: Unterprozess System entwerfen BPMN-Diagramm	27
Information 9: Unterprozess System implementieren Beschreibung.....	28
Information 10: Unterprozess System implementieren BPMN-Diagramm	29
Information 11: Unterprozess Projektteam koordinieren BPMN-Diagramm	30
Information 12: Unterprozess Stundenkonten führen Beschreibung.....	30
Information 13: Unterprozess Stundenkonten führen BPMN-Diagramm	31
Information 14: Unterprozess Teamtreffen durchführen Beschreibung	31

Information 15: Unterprozess Teamtreffen durchführen BPMN-Diagramm	32
Information 16: Unterprozess Aufgaben verwalten Beschreibung	33
Information 17: Unterprozess Aufgaben verwalten BPMN-Diagramm	33
Information 18: Zusammenführen der Ergebnisse	34
Information 19: Zusammenwirken der Werkzeuge	35
Information 20: Teamarbeit!? Zwei Szenarien	36
Information 21: Aktivität System analysieren	37
Information 22: Der LDocEditor GUI-Prototyp in Visual Studio .NET 2003	38
Information 23: Aktivität System entwerfen	39
Information 24: Sequenzdiagramm im EA	40
Information 25: Versionskontrolle im EA mit dem SCC SVN proxy	42
Information 26: Aktivität System implementieren	43
Information 27: LDocEditor API-Doc in Firefox	44
Information 28: LDocEditor API-Doc in Visual Studio	45
Information 29: WebDav-Ansicht im Firefox-Browser	46
Information 30: TortoiseSVN	47
Information 31: AnkhSVN	47
Information 32: Aktivität Aufgaben verwalten	48
Information 33: <i>Todo</i> -Listenverwaltung in Basecamp	48
Information 34: Phasen, Werkzeuge und die Produktdokumentation	50

6.6 Literatur

Hier werden die im Rahmen dieser Studienarbeit als Quellen verwendeten Dokumente aufgelistet. Alle Dokumente, bei denen dies möglich war, wurden mit dieser Arbeit auf CD ausgeliefert und sind in der Tabelle verlinkt.

[BPMN1.0]	Business Process Management Initiative (BPMI): Business Process Modeling Notation (BPMN), Version 1.0, BPMI.org, May 2004. BPMN1.0 BPMN-v1.0-May-3-2004.pdf
[C&M-I-SM]	Cooperation & Management: SYSTEMMODELLIERUNG, Kurseinheit zu INTERNET-SYSTEME UND WEB-APPLIKATIONEN, http://www.cm-tm.uka.de/iswa , Universität Karlsruhe, C&M (Prof. Abeck). C&M-I-SM systemmodellierung 04-11-07-krutz-2.doc C&M-I-SM systemmodellierung 04-11-07-krutz.ppt
[C&M-P-AW]	Cooperation & Management, C&M-ARBEITSWEISE, http://www.cm-tm.uka.de/extern , Ordner "Profil", Universität Karlsruhe (TH), C&M (Prof. Abeck). C&M-P-AW cm-arbeitsweise 04-11-22-abeck kom.doc C&M-P-AW cm-arbeitsweise 04-11-03-abeck.doc C&M-P-AW cm-arbeitsweise 04-11-03-abeck.ppt
[C&M-P-KE]	Cooperation & Management, C&M-KURSENTWICKLUNG, http://www.cm-tm.uka.de/extern , Ordner "Profil", Universität Karlsruhe (TH), C&M (Prof. Abeck). C&M-P-KE cm-kursentwicklung 04-09-16-abeck.doc C&M-P-KE cm-kursentwicklung 04-09-16-abeck.ppt

- [C&M-P-SE] Cooperation & Management, C&M-SOFTWAREENTWICKLUNG, <http://www.cm-tm.uka.de/extern>, Ordner "Profil", Universität Karlsruhe (TH), C&M (Prof. Abeck).
[C&M-P-SE_cm-softwareentwicklung_04-11-03-abeck.doc](#)
[C&M-P-SE_cm-softwareentwicklung_04-11-03-abeck.ppt](#)
- [Oe01] Bernd Oestereich: Objektorientierte Softwareentwicklung : Analyse und Design mit der Unified Modeling Language; Oldenbourg-Verlag, 2001.
- [Oe04] Bernd Oestereich: [Objektorientierte Softwareentwicklung: Analyse und Design mit der UML 2.0](#), Oldenbourg-Verlag, 2004.
- [OR03] Martin Owen, Jog Ray: BPMN and Business Process Management – Introduction to the New Business Modeling Standard, Popkin Software 2003.
[OR03 BPMN-and-BPM.pdf](#)
- [Wh04a] Stephen A. White: *Process Modeling Notations and Workflow Patterns*, IBM,
<http://www.bpmn.org/Documents/Notations%20and%20Workflow%20Patterns.pdf>, Januar, 2004.
- [Wh04b] Stephen A. White: *Introduction to BPMN*, IBM,
<http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>, Mai, 2004.
- [CF+04] Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato: *Version Control with Subversion 1.1* Rev12923, 559 Nathan Abbott Way, Stanford, California, 94305, USA, Creative Commons, <http://svnbook.red-bean.com/>, 2004.